Isabelle/HOL 中物理量、单位和测量的自动推理

Simon
Foster约克大学

Burkhart Wolff巴
黎-萨克雷大学

arXiv:2302.07629v1 [cs.LO] 15 Feb 2023

摘要——**对信息物理系统和机器人系统进行形式化验证，要求我们能够准确地建模现实世界中存在的物理量。在此类量中显式标注单位可以提高严谨性，因为这样我们就能确保计算中各量的兼容性。同时，单位使用不当会危及安全，因此对物理计算进行自动合理性检查十分必要。本文对国际量制（ISQ）及其相关的国际单位制在 Isabelle/HOL 中进行了形式化。我们展示了如何利用 Isabelle 为物理量提供类型系统和自动化证明支持。量由量纲类型参数化，这些类型对应于基向量，因此只有量纲相同的量才能相等。由于底层的“量代数”会在量类型和国际单位制类型上诱导出同余关系，我们开发了专门的策略支持来处理这些关系。我们通过一个测试集验证了这一构造，其中包含量之间以及国际单位制单位之间已知的等价关系。此外，所提出的理论还可用于在国际单位制与其他单位制（如英制单位制（BIS））之间进行类型安全转换。**

一、引言

信息物理系统利用软件控制与物理世界的交互，因此必须考虑物理现象中可量化的属性。现代物理学使用质量、长度、时间和电流等量来描述这类物理属性。这些量通过代数关系与速度、力和能量等导出概念相联系。后者使人们能够对物理方程进行量纲分析，而量纲分析是牛顿物理学的基石。与此同时，物理学家开创了一个名为“计量学”的研究领域，用于科学研究物理量的测量。将计量学融入软件工程至关重要，这能确保物理组件以安全且可预测的方式运行 [1], [2], [3]。

量与测量的国际标准由国际计量局（BIPM）发布，该机构还提供《国际计量学词汇》（VIM）。VIM 实际上定义了两个体系：国际量制（ISQ）和国际单位制（SI，源自法语“ Système international d’unités”）。后者也载于《国际单位制手册》 [5]，这是一本定期更新的标准，最近一次更新是在2019年。最后，VIM 还定义了具体的参考测量程序，并规定了测量误差的术语。

国际单位制是对 ISQ 的完善，由一个连贯的单位体系构成，建立在七个基本单位之上，分别是米、千克、秒、安培、开尔文、摩尔、坎德拉，以及一组用于单位名称和单位符号的二十四个词头，例如毫和千，可用于表示单位的倍数和分数。该体系还为 22 个导出单位规定了名称，例如流明和瓦特，用来表示其他常见的物理量。尽管仍存在许多计量单位制，例如英制单位制（BIS），国际单位制仍可视为所有这些单位制事实上的参照标准。

本文的贡献是在 Isabelle/HOL 中对国际量制和国际单位制进行形式化，并将其深度集成到 Isabelle 的序排序多态类型系统中 [6]。我们的目标有两个：（1）对国际量制及其单位本体进行一致的形式化；（2）支持在信息物理系统的形式化规范和模型中使用单位 [7]。我们对物理量的处理使 Isabelle 的类型系统能够检查单位的正确使用。由于量代数会在量类型上诱导同余关系，因此我们开发了专门的策略支持来捕获这些关系。我们通过一个包含量与国际单位制单位之间已知等价关系的测试集验证了这一构造。此外，所提出的理论还可用于在国际单位制与其他单位制（如 BIS）之间进行类型安全转换。

具体而言，我们为量引入了一种新颖的参数化类型 $N[\mathcal{D}, \mathcal{S}]$，其中 N 是数值类型（例如 $\mathbb{Q}$、 $\mathbb{R}$），$\mathcal{D}$是量纲类型（例如**L**、**M**、**T**），而 $\mathcal{S}$ 是所采用的单位制。与此同时，我们还构建了单位的形式本体，可用于测量。于是，我们可以写出具体的量，例如 $20 *_{\mathbb{Q}}$ 米 :: $\mathbb{R}[L$、国际单位制$]$，表示国际单位制中 20 米的测量值（量纲为长度）；以及 $30 *_{\mathbb{Q}}$ 磅 :: $\mathbb{R}[M$ 、 $BIS]$，表示 BIS 中 30 磅的测量值（量纲为质量）。只有量纲和单位制都相同的量才具有可比性，因此 $20 *_{\mathbb{Q}}$ 米 $= 30 *_{\mathbb{Q}}$ 磅会产生类型错误。然而，我们可以在不同单位制之间进行转换，例如 公制转换$(30 *_{\mathbb{Q}}$ 磅$)$ $\approx$ $9.07 *_{\mathbb{Q}}$ 千克。我们的工作运用了 Isabelle/HOL 的多项高级特性来实现量类型系统，而不依赖依赖类型的复杂性。

总而言之，我们的贡献包括：1）将 ISQ 嵌入 Isabelle/HOL，其中包括量纲、量、单位和转换；2）一种由构造保证可靠性的量类型系统，用于检查量纲和量纲强制转换；3）为量猜想提供自动化证明支持；

4）源自国际计量学词汇的单位形式本体 [4] 和《国际单位制手册》 [5], ，可用于规范和模型。

本文的结构如下。在 §II 中，我们简要回顾相关工作，以便将我们的贡献置于背景之中。在 §III 中，我们开始介绍本文的贡献，阐述量纲类型；这些类型采用宇宙构造和基于类型类的刻画。在 §IV中，我们利用量纲实现量类型 $N[\mathcal{D}, \mathcal{S}]$，其中包括自动化证明。在 §V中，我们实现国际单位制，并建立与之相关的单位本体和方程。在 §VI 中，我们介绍不同单位制之间的换算，例如 国际单位制、*CGS*和*BIS*。最后，在 §VII中，我们评估本文工作并总结。我们全部的理论开发成果可在形式证明档案中找到 [8]。

II. 相关工作

软件和形式化规范中对物理量与测量的需求已得到广泛认可 [9],[1], [2], [3]。Burgue˜no 等人 [1], [10]指出，在机器人软件模型中纳入物理量对安全至关重要，并扩展 UML，为量、量纲和单位增加类型。Flater [3] 主张扩展国际单位制标准，将量纲和单位纳入其中，以支持软件计量学。

量类型已在多种主流数值计算系统中实现，例如 MATLAB[1] 和 Mathematica[2], 通常用于支持单位换算、检查单位一致性以及简化量纲。Hall [2]介绍了一个近期推出的 Python 库，该库实现了量并便于进行量纲分析。我们的工作可作为 ISQ 经过验证的实现的基准，尤其是通过 Isabelle 代码生成器实现的版本 [11]。

此外，已有许多针对编程语言直接实现 ISQ 和国际单位制的工作。Kennedy [12], [13] 为$F^\sharp$提出了量纲类型，将其用作参数化数据的一种方式；Garrigue 和 Ly [14]近期也沿着这一思路进行了阐述。这些工作在类ML语言中直接实现了用于量纲和单位的类型系统，而我们的方法则在参数多态性框架和 HOL 框架内，形式化地推导出这种类型推断。因此，与直接实现相比，我们的方法通过构造保证正确性。

Hayes等人 [9]扩展了Z规范语言，以纳入单位、量纲和量。他们的主要创新是增加了一个运算$M \odot D$，它实际上是一个类型构造器，用于构造数值类型为M 、量纲类型为D的量。后者为我们的方法提供了灵感。不过，他们的工作缺少相应的实现，而我们则有效地为嵌入 Isabelle/HOL 的量提供了类型系统。这超出了Z的表达能力。此外，与[9] 不同，我们的量类型通过量纲宇宙构造传达了底层量纲的语义信息，可用于推理。

Aragon [15]探讨量纲和量的代数结构。他将量形式化为称为 q-数，由一个复数和一个表示单位的标签组成。随后，他探讨了单位标签和量的代数性质。文中并未明确刻画量纲，而只涉及单位。尽管如此，由此得到的性质仍为我们的工作提供了基准，例如证明量纲构成一个加法群。

我们的工作为ISQ提供了一种基础性且实用的实现：它精确实现了量演算，同时允许自动检查量纲、提供高效的证明支持并支持代码生成。我们还提供了经过验证的计量单位本体，可用于形式化规范和模型 [7]。据我们所知，迄今为止，证明助手中尚无可与之媲美的ISQ实现。

III. 量纲

本节将对量纲进行形式化建模，后续各节将使用量纲来参数化物理量。

量纲用于区分不同种类的量。例如，10 kg 和 10m 的量大小相同，但由于它们的量纲分别是质量和长度，因此无法比较。在 ISQ 中有七个基本量，包括长度、质量和时间，分别对应七个基本量纲，我们将在本文稍后讨论这些基本量纲。每个基本量纲都用一个符号表示，例如 L、M、T ，量纲则是这些符号的乘积，每个符号都取一个整数次幂。例如，面积量用量纲 L^2，表示，速度量则用 $L{\cdot}T^{-1}$ 表示。由于我们希望支持不同的单位制，因此这里采用基于向量的通用量纲系统。

在类型理论的语境下，量纲可视为物理量的参数。具体而言，我们可以从概念上用量纲对一个量进行参数化；由于相等关系和序关系通常是齐次的，因此只能比较量纲相同的量。然而，要在 Isabelle/HOL 这样的定理证明器中实现这一点并不容易，因为它缺乏依赖类型，所以我们需要借助不同机制来刻画量纲类型。在本例中，我们使用类型类，从而能够有效地选出一组给定的类型构造器，用来定义量纲。不过，我们首先需要刻画一个全集，使这些类型构造器在该全集下封闭。因此，我们先定义一个量纲全集，随后再使用类型类，有效地定义量纲类型与该全集之间的同态。

这一总体方法如图 1所示。我们定义了：（1）量纲全集；（2）一个类型类（*dim-type*），用于从语法上刻画表示量纲的一类类型；（3）定义一组实例化*dim-type*的单元素类型和类型构造器，并可用于参数化量。实际上，这实现了一个基于量纲算术运算符归纳定义的类型族。我们的方法具有通用性，可应用于不同的测量系统，不过目前我们专注于 ISQ。

[1] https://uk.mathworks.com/discovery/dimensional-analysis.html [2] https://reference.wolfram.com/language/ref/Quantity.html

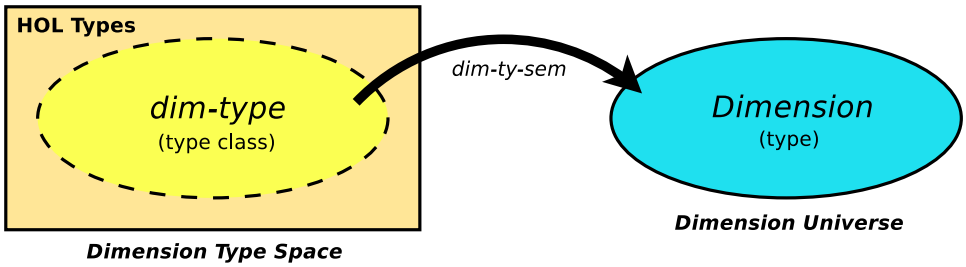

图 1. 将量纲类型映射到量纲全集

A. 量纲全集

我们首先定义量纲全集、用于构造量纲的核心运算符及其性质。假设存在 $n \in \mathbb{N}$ 个基本量，则量纲的形式为 $d_1^{\mathbf{x}_1} \cdot d_2^{\mathbf{x}_2} \cdots d_n^{\mathbf{x}_n}$，即量纲符号（$d_i$）分别取向量 $\mathbf{x}$中的幂的乘积。下面展示了量纲向量 $\mathbf{x}$ 在 Isabelle 中的编码。

定义 1 （量纲向量）。

$$\textbf{\textit{typedef}}\,(N, I)\,dimvec = (UNIV :: (I :: enum) \Rightarrow N)$$

Isabelle/HOL 中的 **typedef** 命令引入一种新类型，该类型由现有类型的一个非空子集刻画。在本例中，我们引入带有两个参数 N 和 I 的新类型 量纲向量。我们选择从 I 到N 的所有全函数构成的集合 $UNIV$ 作为特征集。在后续的类型定义中，我们将使用**typedef** 进一步限制该子集。

从概念上讲，量纲向量就是一个从可枚举索引类型 I（表示可能的量纲）到数值类型 N 的全函数，而 N 至少应构成一个环（例如 $\mathbb{Z}$）。可枚举（$enum$）排序约束（$I :: enum$）要求 I 与一个值列表同构，因此它也是有限类型。在 ISQ 中，例如有 I = {**L**、**M**、**T**、**I**、**Θ**、**N**、**J**}。

接下来，我们定义核心量纲构造器。为简洁起见，我们将其表示为使用 λ-项的函数，但从技术上讲，它们是使用 lifting 软件包定义的 [16]。

定义 2 （量纲构造器）。

$$\mathbf{1} \triangleq (\lambda\, i.\, 0) \quad \boldsymbol{b}(i) \triangleq \mathbf{1}(i \mapsto 1)$$

$$\mathbf{x} \cdot \mathbf{y} \triangleq (\lambda\, i.\, \mathbf{x}(i) + \mathbf{y}(i)) \quad \mathbf{x}^{-1} \triangleq (\lambda\, i.\, -\mathbf{x}(i))$$

这里，$\mathbf{1}$ 表示零量纲，它不对应任何物理量。它可以描述无量纲量，例如数学常数（π、e 等）和函数。

函数 $\mathbf{b}(i)$（其中 $i \in I$）通过将 $\mathbf{1}$ 中 i 的映射更新为幂 1，从基本量 i 构造一个基本量纲。基本量纲的向量中恰好有一个元素映射到 1，其余元素均为 0。我们还定义谓词 $is\text{-}BaseDim :: (N,I)$ 量纲向量 $\Rightarrow \mathbb{B}$，用于确定一个量纲向量是否对应于基本量纲。

两个量纲的乘积（$\mathbf{x} \cdot \mathbf{y}$）只需将所有幂逐项相加，而逆（$\mathbf{x}^{-1}$）则将每个幂取反。现在，我们也可以使用通常的定义来得到除法：$\mathbf{x}/\mathbf{y} \triangleq \mathbf{x} \cdot \mathbf{y}^{-1}$。根据这些定义，我们可以证明以下群定理：

定理 1。 如果 $(N, +, 0, -)$ 构成一个阿贝尔群，那么$((N,I)$量纲向量$, \cdot, \mathbf{1}, ^{-1})$ 也构成一个阿贝尔群。

因此，可以使用阿贝尔群运算律对量纲表达式进行等式改写，而这一过程由 Isabelle 的化简器自动完成。

借助 Isabelle 代码生成器 [11]，我们还可以采用另一种途径，高效地证明量纲相关性质。由于基本量的集合 I 可枚举，我们总能将量纲向量转换为 N 的列表，反之亦然。我们通过函数 $mk\text{-}dimvec$ 实现这一点：它将长度为 $|I|$ 的 N 列表转换为 N 上的量纲向量。

定义 3 （将列表转换为量纲）。

$$mk\text{-}dvec(ds) \triangleq \left(\begin{array}{l} \textbf{if}\ (length(ds) = |I|) \\ \textbf{then}\ (\lambda\, d.\, ds(enum\text{-}ind(d)))\ \textbf{else}\ \mathbf{1} \end{array} \right)$$

由于I 可枚举，每个量纲都可以分配一个自然数，该数同时表示它在底层列表中的位置。函数 $enum\text{-}ind :: (I ::enum) \Rightarrow \mathbb{N}$可提取可枚举类型中某个值的位置索引。对于 ISQ，例如有 $enum\text{-}ind(\mathbf{L}) = 0$ 和 $enum\text{-}ind(\mathbf{T}) = 2$。然后，只需查找列表 ds 中枚举索引对应的值，就可以构造出一个量纲。

所有可能的量纲都可以通过 $mk\text{-}dvec$ 构造，因此我们可以将它用作 Isabelle 代码生成器的所谓“代码数据类型”。随后，量纲会在 SML 或 Haskell 中编码为一种代数数据类型，它只有一个与 $mk\text{-}dvec$ 对应的构造器，例如：

```
datatype ('a, 'b) dimvec = Mk_dimvec of 'a list
```

接下来，我们证明群运算符的代码方程定理。这些同态律能够实现高效执行：

定理 2. 对于量纲向量空间 (N,I) 量纲向量，若$|xs| = |ys| = |I|$，则以下代码方程成立：

$$\begin{aligned} \mathbf{1} &= mk\text{-}dvec\,(replicate\ |I|\ 0) \\ mk\text{-}dvec(xs) \cdot mk\text{-}dvec(ys) &= map\,(\lambda(x,y).\, x + y)\,(zip\, xs\, ys) \\ (mk\text{-}dvec(xs))^n &= mk\text{-}dvec\,(map\,(\lambda\, x.\, n \cdot x)\, xs \\ (mk\text{-}dvec(xs))^{-1} &= mk\text{-}dvec\,(map\,(\lambda\, x.\, -x)\, xs \end{aligned}$$

这些定理为该数据类型上的可执行函数给出了具体的定义方程。零量纲是一个

由 0 个长度幂组成的列表 $[]$。两个等长列表 *xs* 和 *ys* 的乘法，是将每个元素逐对相加。提升到 *n* 次幂会将列表中的每个元素乘以 *n*。取逆幂会将每个元素取负。借助这些方程，我们可以对由列表构造的量纲执行高效的量纲运算。

ISQ 中的量纲使用具体的量纲索引类型 *sdim* 表示：

定义 4 （国际量制基本量）。

$$\textbf{\textit{datatype}}\ sdim = Length \mid Mass \mid Time \mid Current \mid Temperature \mid Amount \mid Intensity$$

类型同义词$Dimension = (\mathbb{Z}, sdim)$ 量纲向量

只需通过类型类实例化证明 *sdim* 是可枚举的，随后便可为 ISQ 中的量纲向量创建特定的类型同义词 *Dimension*。为方便起见，我们接着为每个基本量定义量纲向量，例如 **L** ≜ **b(** 长度 **)**。

B. 量纲类型

定义量纲全集后，下一步是刻画量纲类型这一族。这些量纲类型将用于参数化我们的量，并确保只能比较量纲类型相同的量。我们通过首先引入一个用于量纲类型的类型类，避免依赖类型的需求。

定义 5 （量纲类型类）。

类 *dim-type* = 单元型 +固定

dim-ty-sem :: $\mathcal{D}$自身 ⇒ 量纲

类*basedim-type* = *dim-type* +假设

is-BaseDim: *is-BaseDim*(*QD*($\mathcal{D}$))

类型类用于刻画一族类型，其中每种类型都实现具有特定性质的给定函数签名，例如幺半群和群等代数结构。**class** 命令会引入一个具有给定名称的类型类，并可扩展现有类型类。**fixes** 子命令会在签名中声明一个新的带类型符号，而 **assumes**则会引入签名中这些符号所满足的性质。

dim-type 类型类用于刻画单元型 $\mathcal{D}$（即基数为 1 的类型），并将其关联到特定量纲。类型 $\mathcal{D}$itself 表示 Isabelle/HOL 中作为值的类型。因此，dim-ty-sem 可视为从属于 dim-type 类型类的类型映射到特定量纲的函数，如图 1 所示。我们可以使用语法构造器 TYPE(α)，为特定类型 α 获取一个类型为 αitself 的值。这实际上在值层级和类型层级的量纲之间引入了同构。为方便起见，我们引入记号 QD($\mathcal{D}$) ≜ dim-ty-sem TYPE($\mathcal{D}$)，用于获取给定量纲类型的量纲。basedim-type 类进一步专门化了 dim-type，要求映射得到的量纲是基本量纲。

我们使用这些类来涵盖量纲类型的类型构造器集合。首先，我们构造一些类型来表示基本量纲，并将其定义为单元型类型。例如，我们如下定义长度类型：

typedef *Length* = (*UNIV* :: *unit set*)

它利用了类型定义会从一个集合中生成新的类型名称这一事实（在本例中，该集合只包含 *unit* 类型的唯一元素）。虽然这似乎与定义 *Length* 中引入的构造子 4存在冲突，但这些名称属于不同的命名空间。此处的 *Length* 是一个“标签类型”，其成员不传达信息，而是在语法上表示量纲类型。

我们定义七个这样的类型，分别对应 ISQ 的每个基本量，此外还定义一个名为 **1** 的特殊类型，它对应于无量纲量。每个基本量纲都通过将其映射到上一节中引入的相应量纲符号来实例化 *basedim-type* 类，例如，*QD*(*Length*) = *Length*。

接下来，我们在类型层级上引入量纲的算术运算符。乘积类型构造子和逆类型构造子的定义如下：

类型定义 ($\mathcal{D}_1$::维度类型，$\mathcal{D}_2$::维度类型) 维度乘积 =(全集 :: 单位集)**类型定义** ($\mathcal{D}$::维度类型) 维度逆 =(全集 :: 单位集)

它们同样是标签类型，但参数必须属于 *dim-ty* 类。这确保量纲类型在乘积和求逆运算下封闭。利用这些类型构造子和基本量纲类型，我们可以在类型层级上归纳定义代数量纲。我们为这些类型构造子指定以下 量纲类型语义实现：

定义 6 （量纲类型的语义解释）。

$$\textit{dim-ty-sem}(d :: (\mathcal{D}_1, \mathcal{D}_2)\ \textit{DimTimes}) = QD(\mathcal{D}_1) \cdot QD(\mathcal{D}_2)$$
$$\textit{dim-ty-sem}(d :: (\mathcal{D})\ \textit{DimInv}) = QD(\mathcal{D})^{-1}$$

这些定义将类型构造子与底层量纲运算符联系起来。*DimTimes* 类型的语义会计算各参数$\mathcal{D}_1$ 和 $\mathcal{D}_2$对应的底层值层级量纲，并将它们相乘。*DimInv* 类型的语义同样会计算量纲，然后取其逆。我们为这些类型构造子定义了常用的数学语法，因此可以写出类似 $\mathbf{M} \cdot \mathbf{L}$ 和$\mathbf{T}^{-1}$这样的量纲类型。我们还定义了一个表示除法的类型同义词，即 ($\mathcal{D}_1, \mathcal{D}_2$) *DimDiv* ≜ $\mathcal{D}_1 \cdot \mathcal{D}_2^{-1}$，并赋予它常用的语法。此外，我们还在类型层级上定义了固定数量的幂和负幂，例如 $\mathcal{D}^{-3} = (\mathcal{D} \cdot \mathcal{D} \cdot \mathcal{D})^{-1}$。

现在，我们还可以使用类型同义词，创建 ISQ 中规定的导出量纲集合。例如，我们定义 速度 ≜ $\mathbf{L} \cdot \mathbf{T}^{-1}$ 和 压强 ≜ $\mathbf{L}^{-1} \cdot \mathbf{M} \cdot \mathbf{T}^{-1}$，从而为形式化规范中的量纲使用提供一套术语。图 2还展示了更多示例，并说明了 Isabelle/HOL 中实现的量纲数学语法。

C. 维度规范化

与值层级的量纲不同，语法形式不同的量纲类型彼此不可比较，因为它们是

```
type_synonym Area = "L²"
type_synonym Volume = "L³"
type_synonym Velocity = "L·T⁻¹"
type_synonym Acceleration = "L·T⁻²"
type_synonym Frequency = "T⁻¹"
type_synonym Energy = "L²·M·T⁻²"
type_synonym Power = "L²·M·T⁻³"
type_synonym Force = "L·M·T⁻²"
type_synonym Pressure = "L⁻¹·M·T⁻²"
type_synonym Charge = "I·T"
```

图 2. Isabelle/HOL 中导出的量纲类型表达式

不同的类型表达式。例如，直观上，$\mathbf{L}\cdot\mathbf{T}^{-1}\cdot\mathbf{T} = \mathbf{L}$是一个事实，可以利用群公理证明。然而，在类型层级，$\mathbf{L}\cdot\mathbf{T}^{-1}\cdot\mathbf{T}$和$\mathbf{L}$是不同的类型表达式，而 Isabelle 中没有内置的规范化功能。因此，我们需要在 Isabelle/ML 中为 ISQ 量纲实现自己的规范化函数 $normalise(\mathcal{D})$，以便关联具有不同语法形式的量纲上的量。我们的规范化函数会计算量纲表达式的量纲向量，并据此生成范式。

我们使用 ML 函数 $typ\text{-}to\text{-}dim :: typ \Rightarrow intlist$ 来实现量纲类型求值。该函数会将由基本量纲和量纲算术运算符构成的类型转换为量纲向量列表，表示方式见定义 3。例如，$typ\text{-}to\text{-}dim(\mathbf{L}) = [1,0,0,0,0,0,0]$以及 $typ\text{-}to\text{-}dim(\mathcal{D}^{-1}) = map(\lambda x.\ -x)\,(typ\text{-}to\text{-}dim(x))$。

求得量纲表达式后，我们可以用它构造范式。这是一个有序的量纲表达式，形式为 $\mathbf{L}^{x_1}\cdot\mathbf{M}^{x_2}\cdot\mathbf{T}^{x_3}\cdots\mathbf{J}^{x_7}$，但会省略满足 $\mathbf{x}_i = 0$ 的项。如果所有这类项都为 0，则该函数会生成无量纲量 $\mathbf{1}$。例如，$normalise(\mathbf{T}^4\cdot\mathbf{L}^{-2}\cdot\mathbf{M}^{-1}\cdot\mathbf{I}^2\cdot\mathbf{M})$ 得到量纲类型 $\mathbf{L}^{-2}\cdot\mathbf{T}^4\cdot\mathbf{I}^2$。本文后文将使用此规范化函数，以便在具有不同量纲表达式的量之间进行强制转换。

IV. 物理量与测量

本节将关注量本身。与量纲一样，我们将在值层级和类型层级对量进行建模。我们还将介绍计量系统的概念，它用于指定不同量纲所使用的单位，例如$\mathbf{L}$ 对应米，$\mathbf{T}$对应秒。

A. 量的全集与测量系统

我们将量的全集定义为一个记录，其中包含量的大小和量纲字段。

定义 7（量的全集）。

$$\textbf{\textit{record}}\,(N, I :: enum)\ Quantity = mag :: N \\ dim :: (int, I)\ dimvec$$

量类型由数值类型 N（例如 $\mathbb{Q}$、$\mathbb{R}$）和量纲索引类型I参数化，其中数值类型应构成一个域。量的大小是 N 中的一个数，量纲向量则是定义在I上的向量。现在我们可以指定量上的核心算术运算符。为便于说明，我们使用元组语法（x，$\mathcal{D}$），尽管在 Isabelle 中使用的是记录字段。

定义 8（量的算术运算符）。

$$\begin{aligned} 0 &\triangleq (0, \mathbf{1}) \\ 1 &\triangleq (1, \mathbf{1}) \\ (x,\mathcal{D}_1)\cdot(y,\mathcal{D}_2) &= (x\cdot y, \mathcal{D}_1\cdot\mathcal{D}_2) \\ (x,\mathcal{D})^{-1} &= (x^{-1}, \mathcal{D}^{-1}) \\ (x,\mathcal{D}_1)/(y,\mathcal{D}_2) &= (x/y, \mathcal{D}_1/\mathcal{D}_2) \\ (x,\mathcal{D}) + (y,\mathcal{D}) &= (x+y, \mathcal{D}) \\ (x,\mathcal{D}) - (y,\mathcal{D}) &= (x-y, \mathcal{D}) \\ (x,\mathcal{D}_1) \le (y,\mathcal{D}_2) &\Leftrightarrow (x \le y \wedge \mathcal{D}_1 = \mathcal{D}_2) \end{aligned}$$

算术运算符在 Isabelle/HOL 中是重载的，因此可以合法地出现在这些方程的两边。“0”和“1”这两个量分别被指定为大小为 0 和 1 的无量纲量。乘法、求逆和除法都是全运算，只需在数对的两个分量上分别进行。将两个量相乘时，量值和量纲都需要相乘。例如，$(7, \mathbf{L}\cdot\mathbf{T}^{-1})\cdot(2, \mathbf{T}) = (14, \mathbf{L})$。相反，加法和减法是偏运算，只有两个量的量纲相同时才能进行。在 Isabelle/HOL 中，对于量纲不同的两个量，其加法或减法的值未作规定。最后，量之间的大小关系就是其量值之间的大小关系，但要求两个量的量纲相同。

目前形式化定义的量指定了量纲的形式，但没有指定所采用的单位制。为此，我们扩展 *Quantity*类型，以创建“测量系统”：

定义 9（测量系统）。

$$\textbf{\textit{record}}(N, I :: enum, \mathcal{S} :: \text{单位制})\ \text{测量系统} = (N,I)\ \text{量} + unit\text{-}sys :: \mathcal{S}$$

我们为 量 record 添加一个额外的字段 *unit-sys*，以此对其进行扩展。测量系统是一种量，它通过额外的类型参数 $\mathcal{S}$ 指定所使用的单位制；该参数必须属于类型类 单位制。单位制类型是一种单元型，实际上允许我们为量添加标记。这样，我们便能区分采用不同单位制的量，从而避免不恰当地混用。

例如，国际单位制 标记的存在意味着长度量的基本测量单位是米，而 *BIS* 之类的标记则可能表示长度以码为单位。稍后，我们将利用这些标记，实现不同单位制之间的类型安全转换。

所有算术运算符都可以直接提升到测量系统上。由于所有这类函数都是单态的（例如，类型为 $\alpha \Rightarrow \alpha \Rightarrow \alpha$），因此从构造上避免了不同系统的混用。

B. 维度类型化量

定义了量的全集后，接下来我们将类型级维度纳入这一表示中。为求简便，

我们假设所有这类量都附带一个计量系统。此外，我们重点关注量纲属于 ISQ 的量。

定义 10 (量类型)。

$$\textit{typedef}\,(N, \mathcal{D}::\textit{dim-type}, \mathcal{S}::\text{单位制})\,\textit{QuantT} = \{x :: (N, \textit{sdim}, \mathcal{S})\ \text{测量系统}.\ \textit{dim}(x) = \textit{QD}(\mathcal{D})\}$$

$(N, \mathcal{D}, \mathcal{S})$ *QuantT* 类型表示一个量，其数值类型为 N，量纲类型为 $\mathcal{D}$，单位制为 $\mathcal{S}$。该类型定义引入了一个不变式，要求底层量 x 的量纲与量纲类型中指定的量纲一致。在此层次上，我们使用 *sdim* 作为量纲的具体解释，因为 *dim-type* 类要求如此。为方便起见，我们引入类型语法 $N[\mathcal{D}, \mathcal{S}]$，用来表示 $(N, \mathcal{D}, \mathcal{S})$ *QuantT*。我们的 **typedef** 还定义了两个函数，用于在带类型的量与无类型的量之间转换：$\textit{fromQ} :: N[\mathcal{D}, \mathcal{S}] \Rightarrow (N, \textit{sdim}, \mathcal{S})$ 测量系统，以及反向转换的 *toQ*。

将算术运算符 $x + y$ 和 $x - y$ 提升到带类型的量上非常直接，因为它们是单态的，且仅在 x 和 y 的量纲一致时才有定义。因此，我们很容易证明带类型的量构成加法阿贝尔群。我们还定义了数乘 $\textit{scaleQ} :: N \Rightarrow N[\mathcal{D}, \mathcal{S}] \Rightarrow N[\mathcal{D}, \mathcal{S}]$，记作 $n *_{\mathbb{Q}} x$，它用给定的数缩放一个量而不改变其量纲。由此可以证明，带类型的量构成加法阿贝尔群和实向量空间，其中 $(*_{\mathbb{Q}})$ 是数乘运算符。

处理一般的乘法和除法时，情况会更加复杂，因为这些运算需要在类型层级执行量纲运算。例如，如果我们有量 $x :: \mathbb{R}[\mathbf{I}, \text{国际单位制}]$ 和 $y :: \mathbb{R}[\mathbf{T}, \text{国际单位制}]$，那么对 x 和 y 进行乘法是有定义的，其类型应为 $\mathbb{R}[\mathbf{I} \cdot \mathbf{T}, \text{国际单位制}]$。因此，我们引入了专门的函数：*qtimes*、*qinverse* 和 *qdivide*。首先给出这些函数的类型：

$$\begin{aligned} \textit{qtimes} &:: N[\mathcal{D}_1, \mathcal{S}] \Rightarrow N[\mathcal{D}_2, \mathcal{S}] \Rightarrow N[\mathcal{D}_1 \cdot \mathcal{D}_2, \mathcal{S}] \\ \textit{qinverse} &:: N[\mathcal{D}, \mathcal{S}] \Rightarrow N[\mathcal{D}^{-1}, \mathcal{S}] \\ \textit{qdivide} &:: N[\mathcal{D}_1, \mathcal{S}] \Rightarrow N[\mathcal{D}_2, \mathcal{S}] \Rightarrow N[\mathcal{D}_1 / \mathcal{D}_2, \mathcal{S}] \end{aligned}$$

第一个函数将两个具有相同计量系统的量相乘，并使用第 III-B 节中介绍的类型构造子对量纲类型进行“相乘”。严格来说，这里并未进行乘法计算，而是插入了一个表示乘法的类型构造器。同样，*qin-verse* 表示参数化量纲的逆，而 *qdivide* 表示除法。这里实现的效果类似于依赖类型，不过我们需要额外的机制来规范化量纲类型（参见第 III-C 节）。

通过简单地将定义 *qtimes* 和 *qinverse* 提升自定义 8 中量上对应的函数即可得到它们的定义8，具体来说，这是通过使用 *lifting* 软件包 [16] 实现的。为此，我们需要证明 *QuantT* 类型的不变式成立，这需要证明带类型的量族在这两个函数作用下是封闭的。对于 *qmult*，我们需要证明只要 $\dim(x) = \textit{QD}(\mathcal{D}_1)$ 且 $\dim(y) = \textit{QD}(\mathcal{D}_2)$，就有 $\dim(x \cdot y) = \textit{QD}(\mathcal{D}_1 \cdot \mathcal{D}_2)$；这可直接由定义 6 和定义 8 得出。为方便起见，我们采用 $x \bullet y$、x^{-1} 和 x/y 这些常用记法来表示这些函数，但在 Isabelle 中，我们将运算符加粗，以便在语法上加以区分。借助 qtimes 和 qinverse，我们还可以定义正幂和负幂，例如 $x^{-2} = (x \bullet x)^{-1}$。

HOL 中的相等关系（$x = y$）是类型为 $\alpha \to \alpha \to \mathbb{B}$ 的齐次函数；因此，它不能用于比较不同类型的对象。因此，它不能用于比较量纲类型语法形式不同的量（例如 $\mathbf{L} \cdot \mathbf{T}^{-1} \cdot \mathbf{T}$ 和 $\mathbf{L}$）。这促使我们为量定义异构（不）等式：

$$\begin{aligned} \textit{qequiv} &:: N[\mathcal{D}_1, \mathcal{S}] \Rightarrow N[\mathcal{D}_2, \mathcal{S}] \Rightarrow \mathbb{B} \\ \textit{qless-eq} &:: N[\mathcal{D}_1, \mathcal{S}] \Rightarrow N[\mathcal{D}_2, \mathcal{S}] \Rightarrow \mathbb{B} \end{aligned}$$

这些函数只是通过提升底层量上的函数（$=$）和（$\le$）来定义的。它们忽略量纲类型，但根据 §IV-A 中的定义，底层量纲仍必须相同。我们分别用 $x \cong y$ 和 $x \lesssim y$ 表示这两个函数。关系（$\cong$）构成一个等价关系，而（$\lesssim$）构成一个预序。此外，（$\cong$）是（$\bullet$）、（$^{-1}$）和（$*_{\mathbb{Q}}$）的同余关系。

C. 证明支持

我们为带类型量的（不）等式实现了一种基于解释的证明策略，使我们能够将一个猜想拆分为两部分：(1) 量值相等；(2) 维度等价性。这一策略由函数 $\textit{magQ} :: N[\mathcal{D}, \mathcal{S}] \Rightarrow N$ 提供支持，其语法为 $[\![-]\!]_{\mathcal{Q}}$，该函数从带类型的量中提取量值。我们可以使用解释律计算量值，例如下面这些：

$$[\![x + y]\!]_{\mathcal{Q}} = [\![x]\!]_{\mathcal{Q}} + [\![y]\!]_{\mathcal{Q}} \qquad [\![x \bullet y]\!]_{\mathcal{Q}} = [\![x]\!]_{\mathcal{Q}} \cdot [\![y]\!]_{\mathcal{Q}}$$

这类定律直接源自定义 8 中量运算符的定义8。加法的等式隐含地利用了这样一个事实：x 和 y 具有相同的量纲，因此加法在量的全集中是良定义的。接下来，对于两个具有相同（语法）量纲的量，我们有以下传递定理：

定理 3 (量传递律)。

$$x = y \Leftrightarrow ([\![x]\!]_{\mathcal{Q}} = [\![y]\!]_{\mathcal{Q}}) \qquad x \le y \Leftrightarrow ([\![x]\!]_{\mathcal{Q}} \le [\![y]\!]_{\mathcal{Q}})$$

在这两种情况下，我们无需检查量纲的等价性，因为根据构造，我们知道 x 和 y 具有相同的类型，因此也具有相同的量纲。只需检查底层量值是否满足该关系即可。对于我们的异构（不）等式关系，有以下传递定理：

定理 4 (异构传递律)。给定量 $x :: N[\mathcal{D}_1, \mathcal{S}]$ 和 $y :: N[\mathcal{D}_2, \mathcal{S}]$，我们有

$$x \cong y \Leftrightarrow ([\![x]\!]_{\mathcal{Q}} = [\![y]\!]_{\mathcal{Q}} \wedge \textit{QD}(\mathcal{D}_1) = \textit{QD}(\mathcal{D}_2)).$$

然后，我们可以通过计算底层的量值和量纲，并利用

数值和维度定律。我们提供了一种名为 *si-simp* 的证明方法，它使用化简器执行转移和解释，并额外调用域化简定律。另一种名为 *si-calc* 的方法也会使用代码生成器编译量纲向量（参见定义3），因此能够高效地证明维度等式。例如，我们可以自动证明以下代数定律：

定理5 （量的代数定律）。

$$a *_Q (x + y) = (a *_Q x) + (a *_Q y)$$
$$x \bullet y \cong y \bullet x$$
$$(x \bullet y)^{-1} \cong x^{-1} \bullet y^{-1}$$

D. 强制转换与维度规范化

通过强制转换在同一量纲的两种语法表示之间进行转换，可以避免使用异构数量关系（$\cong, \lesssim$）。此外，我们可以利用 Isabelle 精巧的语法和检查管线来规范化量纲，从而自动将量强制转换为范式。这提高了库的易用性，因为可以直接使用常见的关系（$=$）和（$\leq$）。

我们实现了一个函数 $dnorm :: N[\mathcal{D}_1, \mathcal{S}] \Rightarrow N[\mathcal{D}_2, \mathcal{S}]$，它可以在具有不同量纲形式的量之间进行转换。为了有效使用该函数，必须预先知道目标量纲 $\mathcal{D}_2$。其定义如下：

$$dnorm(x) \triangleq (\textbf{若 } QD(\mathcal{D}_1) = QD(\mathcal{D}_2) \textbf{ 则 } toQ\,(fromQ(x)) \textbf{否则 } 0)$$

该函数会检查源维度和目标维度（$\mathcal{D}_1$ 和 $\mathcal{D}_2$）是否相同。如果相同，则通过*fromQ* 擦除类型，并使用 *toQ*重新赋予新的量纲类型，从而执行强制转换。否则，它将返回一个目标量纲有效、但大小为 0 的量。例如，若有 $x :: \mathbb{R}[\mathbf{L} \cdot \mathbf{T}^{-1} \cdot \mathbf{T}, SI]$，，则可以使用$dnorm(x) :: \mathbb{R}[\mathbf{L}, SI]$，得到一个具有等价维度的量，因为 $QD(\mathbf{L} \cdot \mathbf{T}^{-1} \cdot \mathbf{T}) = QD(\mathbf{L})$。一般而言，对于两个等价量 $x \cong y$，有 $dnorm(x) = y$。

接下来，我们扩展 Isabelle 的检查管线，使其支持量纲规范化，从而可以自动计算 $\mathcal{D}_2$。我们通过实现一个 SML 函数 *check-quant* 来完成这一点；该函数接收一个项，并为其添加维度信息。每当它遇到 $dnorm(t)$ 的实例时，就提取 t 的类型，该类型应为 $N[\mathcal{D}, \mathcal{S}]$。在此情况下，我们将 *dnorm* 的实例丰富为具有类型$N[\mathcal{D}, \mathcal{S}] \Rightarrow N[normalise(\mathcal{D}), \mathcal{S}]$。

然后，我们将 *check-quant* 插入 Isabelle 的项检查管线中。从技术上讲，这是通过一个名为 Syntax Phases.term check 的 Isabelle/ML API 函数实现的，该函数允许我们在项检查过程中添加一个新阶段。在本例中，我们将其添加到类型推断完成之后，以便能够使用未规范化维度类型表达式作为 *check-quant* 的输入。

这一变换的健全性并不依赖于*normalise*的正确性，因为如果计算出的量纲不正确，*dnorm* 将返回 0。不过，其效果类似于实现了依赖类型，但所处的却是一阶多态类型系统。

V. 单位制与国际单位制

本节将一般性地实现单位，尤其是 SI单位制。随后，我们将实现一个源自国际计量学词汇标准 [4]和《国际单位制手册》 [5]的导出单位和词头形式本体。我们的本体由 7 个基本单位、32 个导出单位和认可单位，以及 24 个单位前缀组成。

国际单位制单位只是在 ISQ 中大小为 1 的一个量，通常与一个大小结合使用，以描述一次测量。对于特定单位制而言，一个基本单位 是一个量，其量纲是基本量纲之一。基本单位由谓词 $is\text{-}base\text{-}unit :: N[\mathcal{D}, \mathcal{S}] \Rightarrow \mathbb{B}$ 描述，其定义为 $is\text{-}base\text{-}unit(x) \triangleq (mag(x) = 1 \wedge is\text{-}BaseDim)$。我们引入构造器 $BUNIT(\mathcal{D}, \mathcal{S})$，它使用系统 $\mathcal{S}$ 中的基本量纲类型 $\mathcal{D}$ 构造一个基本单位，其中 $[\![BUNIT(\mathcal{D}, \mathcal{S})]\!]_Q = 1$。

对于国际单位制，我们创建一个单元型类型 *SI*，并实例化 *unit-system*类。然后定义国际单位制的 7 个基本单位：

定义 11 (国际单位制基本单位)。

$$metre \triangleq BUNIT(L, SI) \qquad kilogram \triangleq BUNIT(M, SI)$$
$$ampere \triangleq BUNIT(I, SI) \qquad kelvin \triangleq BUNIT(\Theta, SI)$$
$$mole \triangleq BUNIT(N, SI) \qquad candela \triangleq BUNIT(J, SI)$$

由于秒 常被用作时间单位，我们将其定义为多态基本单位，使它能够有效地存在于多个单位制中。为方便起见，我们创建类型同义词，以便在类型层级指定单位，例如 N 米 $\triangleq N[$长度,国际单位制$]$,，表示国际单位制中量纲为长度的量。

现在，我们终于具备了使用国际单位制单位书写量的条件。在基本层面上，我们可以写出诸如 $20 *_Q$ 米 这样的量，它表示缩放因子为 20 的米 单位，其推断出的类型为 $\mathbb{R}[L, SI]$。我们还可以写出复合单位，例如 $10 *_Q$(米 $\bullet$ 秒$^{-1}$)，其推断出的类型为 $\mathbb{R}[L \cdot T^{-1}]$。我们还可以使用*si-calc* 证明策略证明单位等式，例如 (米 $\bullet$秒$^{-1}$) $\bullet$秒 $\cong$ 米，如下所示：

```
lemma "(metre · second⁻¹) · second ≅Q metre"
  by si_calc
```

类似地，我们可以使用强制转换来证明如下猜想：$dnorm(((5 *_Q($米/秒$)) \bullet (10 *_Q$秒$)) = 50 *_Q$米。

现在，我们可以着手在 Isabelle/HOL 中构建一个取自国际计量学词汇和《国际单位制手册》 [5,页 137]的派生国际单位制单位形式本体：

定义 12 (核心导出单位)。

$$hertz \triangleq second^{-1}$$
$$radian \triangleq metre \bullet metre^{-1}$$
$$steradian \triangleq metre^2 \bullet metre^{-2}$$
$$joule \triangleq kilogram \bullet metre^2 \bullet second^{-2}$$
$$watt \triangleq kilogram \bullet metre^2 \bullet second^{-3}$$
$$coulomb \triangleq ampere \bullet second$$
$$lumen \triangleq candela \bullet steradian$$

Isabelle 可以推断每个此类单位的量纲类型，例如 瓦特 的量纲为 $\mathbf{M} \cdot \mathbf{L}^2 \cdot \mathbf{T}^{-3}$。弧度和球面度的量纲为 $\mathbf{L} \cdot \mathbf{L}^{-1}$ 和 $\mathbf{L}^2 \cdot \mathbf{L}^{-2}$，它们属于不同的量纲类型，但在语义上都等于无量纲量 **1**。有趣的是，其他地方曾有人主张应另设一种角度量纲 [17], [3], ，以便在形式上区分它们。尽管如此，我们选择按照国际单位制的定义来实现，不过未来仍可扩展。

国际单位制定义了 24 个词头，可用于缩放国际单位制单位。下面列出其中一部分：

定义 13 (国际单位制词头)。

$$hecto \triangleq 10^2 \quad kilo \triangleq 10^3 \quad mega \triangleq 10^6 \quad giga \triangleq 10^9$$
$$deci \triangleq 10^{-1} \quad centi \triangleq 10^{-2} \quad milli \triangleq 10^{-3} \quad micro \triangleq 10^{-6}$$

词头不是量，而只是 N 中的抽象数，可用于缩放单位。例如，我们可以写出如下量： $40 *_Q$ 毫 $*_Q$ 米。

国际单位制中还有“认可”单位的概念 [5,页 145],这类量常被用作单位，但严格来说并不属于国际单位制，因为它们的大小不为 1。下面列出其中一部分：

定义 14 (认可的非国际单位制单位)。

$$minute \triangleq 60 *_Q second \quad hour \triangleq 60 *_Q minute$$
$$day \triangleq 24 *_Q hour \quad degree \triangleq (\pi/180) *_Q radian$$
$$litre \triangleq 1/1000 *_Q metre^3 \quad tonne = 10^3 *_Q kilogram$$

在我们的形式化实现中，这些量可以很方便地作为单位处理，不过这类量的类型并不能反映其单位。例如，单位 日、小时 和年 的量纲都为 **T**，正如预期，这意味着它们可以相互比较。因此，我们可以证明单位等式定理，例如$1 *_Q$ 小时 $= 3600 *_Q$ 秒、 $1 *_Q$ 日 $= 86400 *_Q$ 秒以及$1 *_Q$ 公顷 $= 1 *_Q$(百 $*_Q$ 米)^2，并使用 *si-simp* 方法进行化简，这些定理可作为单位换算的基础。类似地，我们可以使用词头来表示导出量之间的关系，例如 $25 *_Q$ 米/秒 $= 90 *_Q$(千 $*_Q$ 米)/小时。

国际单位制单位是根据 7 个物理常数的精确值定义的 [5, 第 127] 页。我们在 Isabelle 中定义如下：

定义 15 （国际单位制定义常数）。

$$\Delta v_{Cs} = 9192631770 *_Q hertz$$
$$\mathbf{c} = 299792458 *_Q (metre \bullet second^{-1})$$
$$\mathbf{h} = (6.62607015 \cdot 10^{-34}) *_Q (joule \bullet second)$$
$$\mathbf{e} = (1.602176634 \cdot 10^{-19}) *_Q coulomb$$
$$\mathbf{k} = (1.380649 \cdot 10^{-23}) *_Q (joule/kelvin)$$
$$N_A = 6.02214076 \cdot 10^{23} *_Q (mole^{-1})$$
$$K_{cd} = 683 *_Q (lumen/watt)$$

Δv_{Cs} 是铯-133 原子的超精细跃迁频率。常数**c** 是真空中的光速，**h** 是普朗克常数。常数**e** 是基本电荷，且 **k** 是玻尔兹曼常数。N_A 是阿伏伽德罗常数。K_{cd} 是频率为 $540 \cdot 10^{12} Hz$的单色辐射的光视效能。这些物理常数为使用特定国际单位制单位进行测量提供了依据。借助这些常数，我们可以整理它们的定义方程，以验证每个单位的定义定理，如下所示：

定理 6 （基础等式）。

$$second \cong (9192631770 *_Q \mathbf{1})/\Delta v_{Cs}$$
$$metre \cong (\mathbf{c}/(299792458 *_Q \mathbf{1})) \bullet second$$
$$kilogram \cong (\mathbf{h}/(6.62607015 \cdot 10^{-34}) *_Q \mathbf{1}) \bullet metre^{-2} \bullet second$$

秒 等于铯原子 ^{133}Cs 辐射的 9192631770 个周期的持续时间。米 是光在 1/299792458 秒内传播的长度。对于千克，该方程实际上定义了单位 $\mathrm{kgm\ s^{-1}}$，然后应用单位 $\mathrm{m^{-2}s}$ 得到量纲为**M**的量。每个方程都使用*si-calc*证明，以此验证我们对国际单位制的实现。最后，我们补全导出单位本体 [5,第 137]页：

定义 16 （进一步导出单位（选录））。

$$newton \triangleq kilogram \bullet metre \bullet second^{-2}$$
$$pascal \triangleq kilogram \bullet metre^{-1} \bullet second^{-2}$$
$$volt \triangleq kilogram \bullet metre^2 \bullet second^{-3} \bullet ampere^{-1}$$
$$farad \triangleq kilogram \bullet metre^{-2} \bullet second^4 \bullet ampere^2$$
$$ohm \triangleq kilogram \bullet metre^2 \bullet second^{-3} \bullet ampere^{-2}$$

此外，国际单位制中的温度以开尔文定义，但通常更习惯用摄氏度表示温度。因此，我们定义$T\ °\mathbf{C} \triangleq (T + 273.15) *_Q$ 开尔文，其中 273.15 是水的冰点。我们可以证明相应的单位等式，它们展示了国际单位制单位之间的等价关系：

定理 7 (导出单位等价关系)。

$$joule \cong newton \bullet metre \quad watt \cong joule/second$$
$$volt = watt/ampere \quad farad \cong coloumb/volt$$

标准中的其余导出单位均已在 Isabelle 中机械化。最后，作为我们方法的一项应用，我们实现了若干天文单位：

定义 17 (天文单位)。

$$julian\text{-}year \triangleq 365.25 *_Q day$$
$$light\text{-}year \triangleq dnorm(\mathbf{c} \bullet julian\text{-}year)$$
$$astromonical\text{-}unit \triangleq 149597870700 *_Q metre$$
$$parsec \triangleq 648000/\pi *_Q astronomical\text{-}unit$$

光年、天文单位和秒差距都是量纲为 **L** 的量。光年是光在一个儒略年内传播的距离。我们将 **c** 乘以 儒略年 ，并对结果进行归一化，以此定义光年。天文单位是地球与太阳之间的大致距离。秒差距是这样一段距离：从该距离处看，1 个天文单位

所对的角为 1 角秒。借助 Isabelle 的柯西实数刻画，我们可以为秒差距赋予一个精确的数学值，即 π。

VI. 单位换算与非国际单位制

本节介绍单位换算方案，可用于在不同单位系统之间换算量。除了国际单位制之外，其他单位系统如今仍被广泛使用，尤其是其他公制单位系统，例如 CGS（厘米-克-秒），以及英制单位系统，包括美国惯用单位制（USC）和英国英制单位制（BIS）。因此，与这些系统的互操作性仍然十分重要。借助目前的量体系，我们已经可以用国际单位制单位描述英制单位，如下所示：

定义 18 （国际单位制中的英制单位）。

$$yard \triangleq 0.9144 *_Q metre \quad mile \triangleq 1760 *_Q yard$$

$$pound \triangleq 0.4535937 *_Q kilogram \quad stone \triangleq 14 *_Q pound$$

$$pint \triangleq 0.56826125 *_Q litre \quad gallon \triangleq 8 *_Q pint$$

这里，我们定义了国际码和磅这两个长度和质量单位，它们于 1959年获得了精确的公制定义。基于这两个单位，我们可以推导出英里 和 英石等单位。品脱 依据英国于 1995年制定的英制定义标准，加仑也同样如此加仑。随后，这些单位便可照常用于构造量。然而，码、磅和品脱等单位存在一个固有问题：它们会因上下文不同而有多种定义。

虽然国际码为0.9144米，但英制码的定义略有不同，约为0.9143992米。该定义基于对英制标准码的测量。英制标准码是一种实物标准器，于1845年制成，经过严格测试后，于1855年通过议会法案成为官方标准 [18], [19]。随后，标准码于1895年与米制标准进行比较测量，结果长度为0.9143992米[3]。

另一方面，美国惯用单位制的定义又略有不同，约为0.9144018米，并由1866年公制法予以标准化 [20]。此外，BIS和UCS中的体积单位“加仑”定义差异很大，分别为277.421立方英寸和231立方英寸（后者是1707年的“葡萄酒加仑” [19]）；品脱等导出单位也存在类似情况。这些不一致对历史测量值尤其成问题，因为它们更可能采用较旧的定义。

因此，当精确测量至关重要时，有必要明确说明所采用的单位制，并定义不同单位制之间的换算因子。即使对于公制单位制，有时也希望使用不同的单位，例如 CGS 制，其中厘米和克用作基本单位。在这种情况下，我们也希望类型系统能够强制确保测量值之间的兼容性。因此，我们对单位制和换算模式都进行了形式化。

换算模式是由 7 个大于零的有理数组成的元组。每个有理数都编码了 ISQ 各量纲对应的一个换算因子。我们为换算模式定义了一种类型，$\mathcal{S}_1 \Rightarrow_U \mathcal{S}_2$，可用于在单位制 $\mathcal{S}_1$ 和 $\mathcal{S}_2$之间转换量。具体而言，我们在 Isabelle/HOL 中使用记录类型和类型定义来实现换算模式；由于篇幅所限，此处省略其定义。

我们定义恒等转换模式 $id_C :: \mathcal{S} \Rightarrow \mathcal{S}$，其中每个因子的值均为 1。我们可以使用运算符 $C_2 \circ C_1 :: \mathcal{S}_1 \Rightarrow_U \mathcal{S}_3$组合两个转换模式 $C_1 :: \mathcal{S}_1 \Rightarrow_U \mathcal{S}_2$ 和 $C_2 :: \mathcal{S}_2 \Rightarrow_U \mathcal{S}_3$，该运算符将$C_1$ 和 C_2 中的各个换算因子分别相乘。类似地，我们可以使用 $inv_C(C_1) :: \mathcal{S}_2 \Rightarrow_U \mathcal{S}_1$求$C_1$ 的逆，即取每个换算因子的倒数。这些运算符构成了一个简单的换算模式范畴。

我们使用转换模式来定义量转换函数 $qconv :: (\mathcal{S}_1 \Rightarrow_U \mathcal{S}_2) \Rightarrow N[\mathcal{D}, \mathcal{S}_1] \Rightarrow N[\mathcal{D}, \mathcal{S}_2]$，其定义如下：

定义 19 （量转换）。

$$qconv_C(m, \mathbf{d}) = \left(\left(\prod_{1 \le i \le 7} C_i^{\mathbf{d}_i}\right) \cdot m, \mathbf{d}\right)$$

给定一个量（m，$\mathbf{d}$）和一个转换模式 C，$qconv$函数通过将C 中的每个元素乘方，指数取对应的量纲元素$\mathbf{d}_i$，从而计算大小m的转换因子。例如，若要将立方（国际）码转换为立方米，首先需要码到米的转换因子，即 0.9144。然后将该值取 3 次方，得到总转换因子 0.764555。正如预期的那样，此操作不会改变量纲本身。

BIS 是英国非米制的度量衡标准，由英国议会于1824年通过法案确立。它规定码和磅分别为长度和质量的标准单位。我们通过创建类型为*BIS*的单位制来对 BIS 建模，并定义码 $\triangleq$ *BUNIT*(*L*,*BIS*) 和 磅 $\triangleq$ *BUNIT*(*M*,*BIS*)。此外，我们还可以创建导出单位，例如英尺 $\triangleq 1/3 *_Q$ 码、英寸 $\triangleq 1/12 *_Q$ 英尺和加仑 $\triangleq 277.421 *_Q$ 英寸[3]。然后，我们可以正式规定某些量是按照 BIS 进行测量的。

我们可以通过创建合适的换算模式，在国际单位制和 BIS 之间转换量 $BSI :: BIS \Rightarrow_U$ 国际单位制。进行此转换所需的长度和质量因子分别为 0.9143993 和 0.453592338。由于时间以秒为单位进行测量，而其他量纲没有对应的解释，我们在换算模式中将它们设为 1。这样，例如，我们可以使用转换 $qconv_{BSI}(1 *_Q$ 盎司$) \approx 37.8 *_Q$ 克[4]。我们

[3]这种差异很可能是由于码的标准长度随着时间推移逐渐缩短，而这一关键事实后来才被发现。

[4]我们在 Isabelle/HOL 中使用精确有理数运算，但为了便于理解，呈现近似十进制展开。

还可以为 UCS 和 CGS 单位制创建单位制，并为其设置合适的换算因子。

虽然我们可以直接使用不同单位制之间的量换算，但通常以国际单位制作为不同单位制的参考框架会更加方便。事实上，国际单位制的一项关键用途就是解决单位制之间的不匹配问题。因此，我们创建一个类型类来表示度量化：

定义 20 （可度量化单位制）。

类型类*metrifiable* = 单位制+**固定**
convschema :: $\mathcal{S}$自身 $\Rightarrow$ （$\mathcal{S} \Rightarrow_U$ 国际单位制）

若存在从 $\mathcal{S}$ 到国际单位制的换算模式，则单位制 $\mathcal{S}$ 是可度量化的。因此，BIS、UCS、CGS 以及国际单位制本身都是可度量化的。因此，对于任意一对可度量化单位制 $\mathcal{S}_1$ 和 $\mathcal{S}_2$，我们定义一个通用换算函数$QMC_{\mathcal{S}_1 \to \mathcal{S}_2} :: N[\mathcal{D}, \mathcal{S}_1] \Rightarrow N[\mathcal{D}, \mathcal{S}_2]$，通过度量化执行换算。该函数首先使用 $\mathcal{S}_1$ 的换算模式将其转换为国际单位制，然后使用 $\mathcal{S}_2$ 的逆模式，将其从国际单位制转换为 $\mathcal{S}_2$。例如，我们可以证明$QMC_{CGS \to BIS}(12 *_Q$ 厘米) $\approx 4.724 *_Q$ 英寸。因此，我们可以利用 Isabelle 类型系统精确指定测量所采用的单位制，并在多种不同单位制之间无缝转换。

VII. 结论

本文介绍了在 Isabelle/HOL 中对 ISQ 进行的全面形式化。我们的形式化使我们能够精确定义特定系统所采用的量纲、单位和单位制。此外，我们还可以利用 Isabelle 类型系统确保在计算中只能组合具有相同量纲和单位制的测量值。我们介绍了一个规模可观的理论开发成果，其中包含约 2500 行定义和证明，涵盖了国际标准《国际计量学词汇》中规定的 ISQ 和国际单位制 [4]。Isabelle/HOL 形式证明档案中提供的理论，为物理量和测量值提供了一个从构造上来说既可靠又完备的类型系统。鉴于 Isabelle 的类型系统远非简单，我们认为这对于混合系统领域中的应用具有重要意义和实用价值。我们通过检查《国际计量学词汇》和《国际单位制手册》中规定的强制性定义及所描述的推论，对我们的理论进行了验证 [5]。我们实现的早期版本还曾应用于一项工业案例研究，该研究针对自主水下航行器建立了形式模型 [7], ，从而提供了进一步的验证。

未来工作有多个方向。当前使用强制转换处理量纲不匹配的方法，可以通过强制子类型机制 [21]实现更好的自动化。它实际上扩展了类型推断算法，使系统能够通过插入已注册的强制转换函数，自动解决类型不匹配问题。同时，我们在图1中展示的量纲类型刻画方法并非专用于 ISQ，也可以推广到其他通常使用依赖类型解决的问题。例如，我们可以对包含算术运算符的类型表达式进行规范化，以关联由长度参数化的向量。因此，未来我们将研究一种通用方法，利用宇宙构造来证明类型级函数和强制转换的合理性。

参考文献

[1] L. Burgue˜no、T. Mayerhofer、M. Wimmer 和 A. Vallecillo，“在机器人软件模型中使用物理量”，载于《第 1 届机器人软件工程国际研讨会论文集》，RoSE ’18 系列。美国纽约州纽约市：计算机协会，2018 年，第 23–28 页。[2] B. D. Hall，“用于物理量计算的软件”，载于《2020 年 IEEE 工业 4.0 与物联网计量学国际研讨会》，2020 年，第 458–463 页。[3] D. Flater，“软件计量学中的量制”，《测量》，第 168 卷，2021 年。[4] 国际计量局和计量学指南联合委员会，“基本概念和一般概念及相关术语（国际计量学词汇（第 3 版））”。BIPM、JCGM，技术报告，2012 年，2008 年版本，含少量修订。[5] ——，“国际单位制（SI）”，BIPM、J CGM，技术报告，2019 年，第 9 版。[6] T. Nipkow 和 G. Snelting，“通过序排序统一实现类型类和重载解析”，载于《函数式编程语言与计算机体系结构》，第 5 届 ACM 会议，1991 年 8 月 26–30 日，美国马萨诸塞州剑桥，论文集，LNCS 系列，第 523 卷。Springer，1991 年，第 1–14 页。[7] S. Foster、Y. Nemouchi、C. O’Halloran、N. Tudor 和 K. Stephen-son，“Isabelle/SACM 中基于形式模型的保证案例：自主水下航行器案例研究”，载于《软件工程中的形式化方法（FormaliSE 2020）：第 8 届国际会议论文集》。ACM，2020 年。[8] S. Foster 和 B. Wolff，“物理量、单位和测量值的健全类型系统”，形式证明档案，2020 年 10 月，https://isa-afp.org/entries/Physical Quantities.html，形式化证明开发。[9] I. J. Hayes 和 B. P. Mahony，“在形式化规范中使用计量单位”，《计算形式化》，第 7, 3, 329 347, 卷，页码 1995 [ ] 10 1007 –。在线。可访问：https://doi.org/ 。/BF 01211077 [10] ˜ L. Burgueno、T. Mayerhofer、M. Wimmer 和 A. Vallecillo，“在软件模型中规范量”，《信息与软件技术》，第 113 卷，第 82–97 页，2019 年。[11] F. Haftmann 和 T. Nipkow，“通过高阶重写系统实现代码生成”，载于《第 10 届函数式与逻辑编程国际研讨会（ FLOPS ）》，LNCS 系列，第 6009 卷。Springer，2010 年，第 103–117 页。[12] A. Kennedy，“量纲类型”，载于《编程语言与系统——ESOP ’94》，D. Sannella 编。德国柏林、海德堡：Springer Berlin Heidelberg，1994 年，第 348–362 页。[13] ——，“度量单位的类型：理论与实践”，载于《中欧函数式编程学校——第三届暑期学校（CEFP 2009）》，LNCS 系列，Z. Horv´ath、R. Plasmeijer 和 V. Zs´ok 编，第 6299 卷。Springer，2009 年，第 268–305 页。[14] J. Garrigue 和 D. Ly，“类型推断器中的单位”，载于《第 28 届法语应用语言研讨会》，法国 Gourette，2017 年 1 月。 [ ] 01503084 在线。可 [15] 访问：https://hal.archives-ouvertes.fr/hal- S. Aragon，“物理量的代数结构”，《数学化学杂志》，第 31 卷，第 1 期，2004 年 5 月。[16] B. Huffman、和 O. Kuncar，“提升与转移：Isabelle/HOL 中商的模块化设计”，载于 CPP 2013，LNCS 系列，第 8307 卷。Springer，2013 年，第 131–146 页。[17] M. I. Kalinin，“国际单位制（SI）中平面角和立体角的地位”，《计量学》，第 56 卷，第 6 期，2019 年 11 月。[18] P. H. Bigg 和 P. Anderton，“以米为基准的英国码标准”，《英国应用物理学杂志》，第 15 卷，1964 年。[19] R. E. Zupko，《测量革命：科学时代以来的西欧度量衡》。美国哲学学会，1990 年。[20] A. Thompson 和 B. N. Taylor，“国际单位制使用指南”，美国国家标准与技术研究院（NIST），技术报告 811，2008 年。[21] D. Traytel、S. Berghofer 和 T. Nipkow，“扩展 Hindley-Milner 类型推断以支持强制结构子类型”，载于 APLAS 2011，LNCS 系列，第 7078 卷。Springer，2011 年，第 89–104 页。