Towards a Theory of Conversion Relations for Prefixed Units of Measure

- Baltasar Trancón y Widemann[1,2] Markus Lepper[2]

[1]Nordakademie, Elmshorn, DE

[2]semantics gGmbH, Berlin, DE

RAMiCS XX

2023-04-03//06

Agenda

1. Introduction
2. Unit Algebra
3. Conversion Relations
4. Conclusion

Agenda

1. Introduction
2. Unit Algebra
3. Conversion Relations
4. Conclusion

Motivation: A Famous Disaster

Metric Math Mistake Muffed Mars Meteorology Mission

"Nov. 10, 1999: A disaster investigation board reports that NASA's Mars Climate Orbiter burned up in the Martian atmosphere because engineers failed to ***convert units*** *from English to metric. The peer review preliminary findings indicate that one team used English units (e.g. inches, feet and pounds) while the other used metric units for a key spacecraft operation."*

Cost: 328 M$

(Wired, 2010-11-10)

Bottom-Up Evaluation

- Solution is easy in theory:
  - Detect the inconsistency – *pound-force* given, *newton* expected;
  - Multiply by 4.448 221 615 260 5 .

Bottom-Up Evaluation

- Solution is easy in theory:
  - Detect the inconsistency – *pound-force* given, *newton* expected;
  - Multiply by 4.448 221 615 260 5 .
- Support is bad in practice:
  - Many tools (libraries, checkers) available.
  - Current methodology does not specify units in code yet.

Bottom-Up Evaluation

- Solution is easy in theory:
  - Detect the inconsistency – *pound-force* given, *newton* expected;
  - Multiply by 4.448 221 615 260 5 .
- Support is bad in practice:
  - Many tools (libraries, checkers) available.
  - Current methodology does not specify units in code yet.
- Theoretical foundations are deficient.

Top-Down Evaluation

- There is no consensus on requirements:
  - Recent survey [MBBS20] identified 296 libraries and 95 tools (OSS only).
  - Functionalities are not nearly pairwise equivalent.
  - Some are clearly unsound.

Top-Down Evaluation

- There is no consensus on requirements:
  - Recent survey [MBBS20] identified 296 libraries and 95 tools (OSS only).
  - Functionalities are not nearly pairwise equivalent.
  - Some are clearly unsound.
- Nearly all abstractions are **operational**:
  - Including ISO 80000
  - Prescriptive rules for notation, pronounciation, calculation
  - No method for objective justification
  - No distinction between
    - logical necessities,
    - contingent (historical) conventions,
    - outright idiosyncrasies

Our Contributions

- A novel **denotational** approach:
  - Compatible with, but orthogonal to [Ken96]
  - Algebraic–relational formal model of units of measure and their conversion
  - Semantics for *future* tools
  - Operational rules justified by deduction

Our Contributions

- A novel **denotational** approach:
  - Compatible with, but orthogonal to [Ken96]
  - Algebraic–relational formal model of units of measure and their conversion
  - Semantics for *future* tools
  - Operational rules justified by deduction
- Epistemological cleanup:

Abstract	logical necessities
Parameterize	by contingent conventions
Rectify	outright idiosyncrasies

Theoretical Language

- Simple denotational objects:
  - Free abelian groups, direct sums
  - Relations, congruences

Theoretical Language

- Simple denotational objects:
  - Free abelian groups, direct sums
  - Relations, congruences
- Cleanup by virtue of (basic) category theory:

Parameterization functors
Abstraction adjoints, natural transformations
Rectification monads, syntax–semantics distinction

Theoretical Language

- Simple denotational objects:
  - Free abelian groups, direct sums
  - Relations, congruences
- Cleanup by virtue of (basic) category theory:

Parameterization	functors
Abstraction	adjoints, natural transformations
Rectification	monads, syntax–semantics distinction

- Not theory **of** R&A, but theory **by** R&A!

Example Symptoms

1. ISO 80000-1 states that “1 is not a dimension”, but recognizes “1 as a derived unit”.

Example Symptoms

1. ISO 80000-1 states that “1 is not a dimension”, but recognizes “1 as a derived unit”.
2. ISO 80000-1 defines derived units by equations, but equals cannot be substituted for equals in compound unit expressions; e.g. *are*, $\mathrm{a} = \mathrm{dam}^2$, cannot be expanded in the compound *hectare*, ha.

Example Symptoms

1. ISO 80000-1 states that “1 is not a dimension”, but recognizes “1 as a derived unit”.
2. ISO 80000-1 defines derived units by equations, but equals cannot be substituted for equals in compound unit expressions; e.g. *are*, $\mathrm{a} = \mathrm{dam}^2$, cannot be expanded in the compound *hectare*, ha.
3. ISO 80000-1 defines $\mathrm{rad} = \mathrm{m}/\mathrm{m}$ and $\mathrm{sr} = \mathrm{m}^2/\mathrm{m}^2$, but treats them as different from each other, and from 1.

Example Symptoms

1. ISO 80000-1 states that “1 is not a dimension”, but recognizes “1 as a derived unit”.
2. ISO 80000-1 defines derived units by equations, but equals cannot be substituted for equals in compound unit expressions; e.g. *are*, $\mathrm{a} = \mathrm{dam}^2$, cannot be expanded in the compound *hectare*, ha.
3. ISO 80000-1 defines $\mathrm{rad} = \mathrm{m}/\mathrm{m}$ and $\mathrm{sr} = \mathrm{m}^2/\mathrm{m}^2$, but treats them as different from each other, and from 1.
4. The SI prefix families are geometric sequences (welcome *ronna*- and *quetta*- in 2022!), but cannot be written as powers of a generator.

Example Symptoms

1. ISO 80000-1 states that “1 is not a dimension”, but recognizes “1 as a derived unit”.
2. ISO 80000-1 defines derived units by equations, but equals cannot be substituted for equals in compound unit expressions; e.g. *are*, $\mathrm{a} = \mathrm{dam}^2$, cannot be expanded in the compound *hectare*, ha.
3. ISO 80000-1 defines $\mathrm{rad} = \mathrm{m}/\mathrm{m}$ and $\mathrm{sr} = \mathrm{m}^2/\mathrm{m}^2$, but treats them as different from each other, and from 1.
4. The SI prefix families are geometric sequences (welcome *ronna-* and *quetta-* in 2022!), but cannot be written as powers of a generator.
5. Tools tend to specify conversion by way of one canonical unit per dimension. As a result, e.g., *newton-meter* $\leftrightsquigarrow$ *joule*, *gray* $\leftrightsquigarrow$ *sievert*, and even *revolutions-per-minute* $\leftrightsquigarrow$ *becquerel* end up convertible.

Committee Confusion

The 16th Conférence Générale des Poids et Mesures,

considering

- the effort made to introduce SI units into the field of ionizing radiations,
- the risk to human beings of an underestimated radiation dose, a risk that could result from a ***confusion*** between absorbed dose and dose equivalent,
- that the ***proliferation*** of special names represents a danger for the Système International d'Unités and must be avoided in every possible way, but that this rule can be broken when it is a matter of ***safeguarding*** human health,

adopts the special name *sievert*, symbol Sv, for the SI unit of dose equivalent in the field of radioprotection.

[CGPM16.5]

Committee Confusion

The 16th Conférence Générale des Poids et Mesures,

considering

- the effort made to introduce SI units into the field of ionizing radiations,
- the risk to human beings of an underestimated radiation dose, a risk that could result from a ***confusion*** between absorbed dose and dose equivalent,
- that the ***proliferation*** of special names represents a danger for the Système International d'Unités and must be avoided in every possible way, but that this rule can be broken when it is a matter of ***safeguarding*** human health,

adopts the special name *sievert*, symbol Sv, for the SI unit of dose equivalent in the field of radioprotection.

The sievert is ***equal*** to the joule per kilogram.

[CGPM16.5]

Paper Confusion

To convert between measurements in different units of the same dimension, we must specify conversion factors between ***various*** units of that dimension. A natural place to keep this information is in the definition of a unit: each unit specifies how to convert measurements in that unit to measurements in ***any other*** defined unit (for the same dimension).

[All+04]

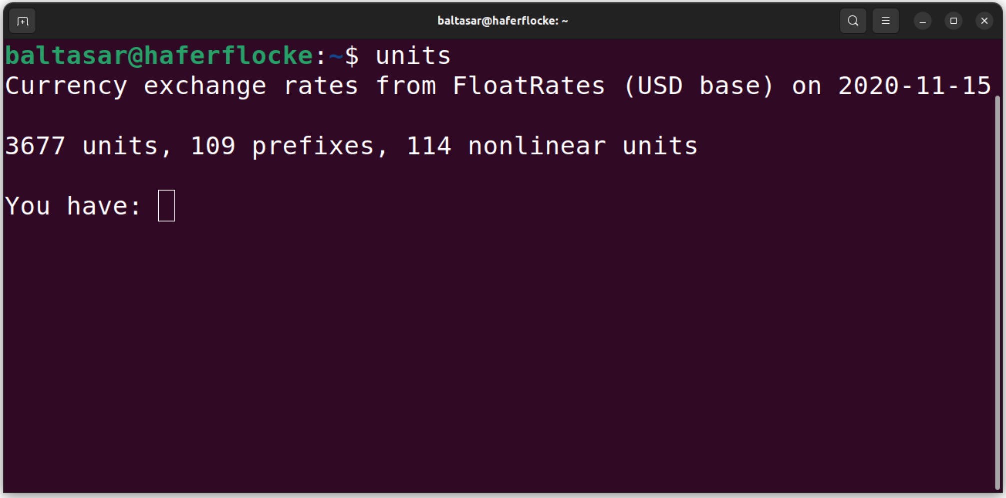
Paper Confusion

To convert between measurements in different units of the same dimension, we must specify conversion factors between ***various*** units of that dimension. A natural place to keep this information is in the definition of a unit: each unit specifies how to convert measurements in that unit to measurements in ***any other*** defined unit (for the same dimension).

Although the number of such conversion factors is quadratic in the number of units, it is not necessary to maintain so many factors explicitly: if we can convert between measurements in units A and B, and between measurements in units B and C, then we can convert between measurements in A and C via B. ***Thus***, it is sufficient to include in the definition of every unit a single conversion factor to a *primary unit* of that dimension, and convert between any two ***commensurable*** units via their common primary unit.

[All+04]

Tool Confusion



Tool Confusion

```
baltasar@haferflocke:~$ units
Currency exchange rates from FloatRates (USD base) on 2020-11-15

3677 units, 109 prefixes, 114 nonlinear units

You have: rpm
You want: becquerel
        * 0.10471976
        / 9.5492966
You have: 
```

Agenda

1. Introduction
2. Unit Algebra
3. Conversion Relations
4. Conclusion

Semantic Sorts

Ratio e.g. $\frac{1000}{1}$, $\frac{5}{9}$, 568.26125, ...

Prefix e.g. k, M, G, ...

Unit e.g. m, N, Hz, ...

Dimension e.g. T, M, I, ...

Free Abelian Groups

- Free abelian group $\mathcal{A}(X)$ over generator set X

Free Abelian Groups

- Free abelian group $\mathcal{A}(X)$ over generator set X
- Represented as finitely supported maps $X \to \mathbb{Z}$
  - Example: $\{a \mapsto 2, b \mapsto 1, c \mapsto -3\}_{/0}$
  - Group operation pointwise additive, ...
  - but commonly written as multiplicative, e.g. $a^2 \cdot b / c^3$
  - Danger of confusion: $1 = \varnothing_{/0} = a^0 b^0 c^0 \cdots$

Free Abelian Groups

- Free abelian group $\mathcal{A}(X)$ over generator set X
- Represented as finitely supported maps $X \to \mathbb{Z}$
  - Example: $\{a \mapsto 2, b \mapsto 1, c \mapsto -3\}_{/0}$
  - Group operation pointwise additive, ...
  - but commonly written as multiplicative, e.g. $a^2 \cdot b / c^3$
  - Danger of confusion: $1 = \varnothing_{/0} = a^0 b^0 c^0 \cdots$
- Functor $\mathcal{A}$ is *free* – left-adjoined to forgetful functor $\mathcal{U} : \mathbf{Ab} \to \mathbf{Set}$

Free Abelian Groups

- Free abelian group $\mathcal{A}(X)$ over generator set X
- Represented as finitely supported maps $X \to \mathbb{Z}$
  - Example: $\{a \mapsto 2, b \mapsto 1, c \mapsto -3\}_{/0}$
  - Group operation pointwise additive, ...
  - but commonly written as multiplicative, e.g. $a^2 \cdot b / c^3$
  - Danger of confusion: $1 = \varnothing_{/0} = a^0 b^0 c^0 \cdots$
- Functor $\mathcal{A}$ is *free* – left-adjoined to forgetful functor $\mathcal{U} : \mathbf{Ab} \to \mathbf{Set}$
  - Unit $\delta_X : X \to \mathcal{U}\mathcal{A}(X)$ — *atomic elements: e. g.* $\delta(a) = a^1$

Free Abelian Groups

- Free abelian group $\mathcal{A}(X)$ over generator set X
- Represented as finitely supported maps $X \to \mathbb{Z}$
  - Example: $\{a \mapsto 2, b \mapsto 1, c \mapsto -3\}_{/0}$
  - Group operation pointwise additive, ...
  - but commonly written as multiplicative, e.g. $a^2 \cdot b / c^3$
  - Danger of confusion: $1 = \varnothing_{/0} = a^0 b^0 c^0 \cdots$
- Functor $\mathcal{A}$ is *free* – left-adjoined to forgetful functor $\mathcal{U} : \mathbf{Ab} \to \mathbf{Set}$
  - Unit $\delta_X : X \to \mathcal{U}\mathcal{A}(X)$ — *atomic elements: e. g.* $\delta(a) = a^1$
  - Counit $\varepsilon_G : \mathcal{A}\mathcal{U}(G) \to G$ — *evaluate group expressions*

Free Abelian Groups

- Free abelian group $\mathcal{A}(X)$ over generator set X
- Represented as finitely supported maps $X \to \mathbb{Z}$
  - Example: $\{a \mapsto 2, b \mapsto 1, c \mapsto -3\}_{/0}$
  - Group operation pointwise additive, . . .
  - but commonly written as multiplicative, e.g. $a^2 \cdot b / c^3$
  - Danger of confusion: $1 = \varnothing_{/0} = a^0 b^0 c^0 \cdots$
- Functor $\mathcal{A}$ is *free* – left-adjoined to forgetful functor $\mathcal{U} : \mathbf{Ab} \to \mathbf{Set}$
  - Unit $\delta_X : X \to \mathcal{U}\mathcal{A}(X)$ — *atomic elements: e. g.* $\delta(a) = a^1$
  - Counit $\varepsilon_G : \mathcal{A}\mathcal{U}(G) \to G$ — *evaluate group expressions*
  - Monad multiplication $\lambda_X : \mathcal{U}\mathcal{A}^2(X) \to \mathcal{U}\mathcal{A}(X)$ — *flatten: e.g.* $\lambda\big((a^3)^2\big) = a^6$

Direct Sum

- Direct sum $G_1 \times G_2$ of pair of abelian groups
  - Cartesian product of carrier sets
  - All operations elementwise

Direct Sum

- Direct sum $G_1 \times G_2$ of pair of abelian groups
  - Cartesian product of carrier sets
  - All operations elementwise
- $\times$ is a *product* — right-adjoined to diagonal functor $\Delta : \mathbf{Ab} \to \mathbf{Ab}^2$

Direct Sum

- Direct sum $G_1 \times G_2$ of pair of abelian groups
  - Cartesian product of carrier sets
  - All operations elementwise

- $\times$ is a *product* — right-adjoined to diagonal functor $\Delta : \mathbf{Ab} \to \mathbf{Ab}^2$
  - Projections $\pi_i : G_1 \times G_2 \to G_i$ — *e.g.* $\pi_1(7, 4) = 7$

Direct Sum

- Direct sum $G_1 \times G_2$ of pair of abelian groups
  - Cartesian product of carrier sets
  - All operations elementwise

- $\times$ is a *product* — right-adjoined to diagonal functor $\Delta : \mathbf{Ab} \to \mathbf{Ab}^2$
  - Projections $\pi_i : G_1 \times G_2 \to G_i$ — *e.g.* $\pi_1(7, 4) = 7$
  - Pairing $\langle f_1, f_2 \rangle : G \to H_1 \times H_2$ from $f_i : G \to H_i$ — *e.g.* $\langle \mathrm{div}, \mathrm{mod} \rangle(7, 4) = (1, 3)$

Direct Sum

- Direct sum $G_1 \times G_2$ of pair of abelian groups
  - Cartesian product of carrier sets
  - All operations elementwise
- $\times$ is a *product* — right-adjoined to diagonal functor $\Delta : \mathbf{Ab} \to \mathbf{Ab}^2$
  - Projections $\pi_i : G_1 \times G_2 \to G_i$ — *e.g.* $\pi_1(7, 4) = 7$
  - Pairing $\langle f_1, f_2 \rangle : G \to H_1 \times H_2$ from $f_i : G \to H_i$ — *e.g.* $\langle \mathrm{div}, \mathrm{mod} \rangle(7, 4) = (1, 3)$
- Fixing G_1 yields a monad (monoid labeling)

$$\eta(x) = (1, x) \qquad \mu\big(a, (b, x)\big) = (ab, x)$$

Composition of Monads

- Create model structures by composing monads:
  - Composition of monads need not be another monad.
  - Sufficient condition: distributive law

Composition of Monads

- Create model structures by composing monads:
  - Composition of monads need not be another monad.
  - Sufficient condition: distributive law
- Canonical homomorphism $\beta_{G,X} : \mathcal{UA}(G \times X) \to G \times \mathcal{UA}(X)$

Composition of Monads

- Create model structures by composing monads:
  - Composition of monads need not be another monad.
  - Sufficient condition: distributive law
- Canonical homomorphism $\beta_{G,X} : \mathcal{UA}(G \times X) \to G \times \mathcal{UA}(X)$
- $(G\times) \circ \mathcal{UA}$ is an adequate model of unit **semantics**.
  - Composite monad by virtue of β

Composition of Monads

- Create model structures by composing monads:
  - Composition of monads need not be another monad.
  - Sufficient condition: distributive law
- Canonical homomorphism $\beta_{G,X} : \mathcal{UA}(G \times X) \to G \times \mathcal{UA}(X)$
- $(G\times) \circ \mathcal{UA}$ is an adequate model of unit **semantics**.
  - Composite monad by virtue of β
- $\mathcal{UA} \circ (G\times)$ is an accurate model of unit **syntax**.
  - Likely not a suitable monad – no group-friendly distributive law exists;
  - but maps naturally to the former by virtue of β.
  - Still useful: composite unit $\lfloor\ \rfloor = \delta\eta$

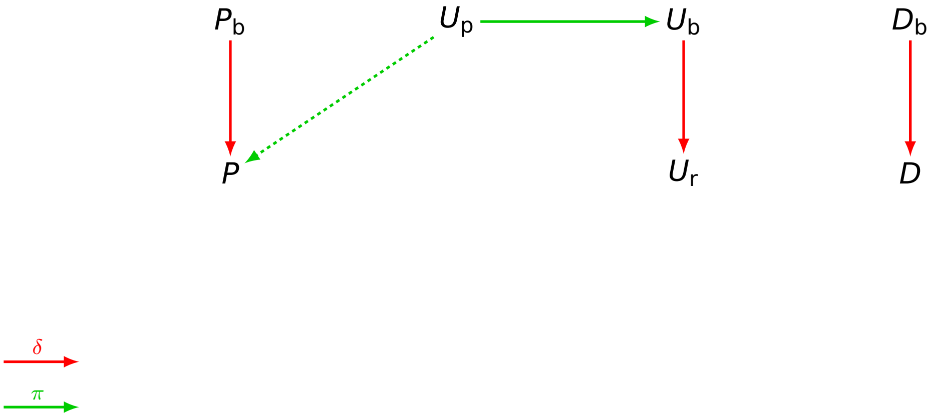
Big Picture

Big Picture

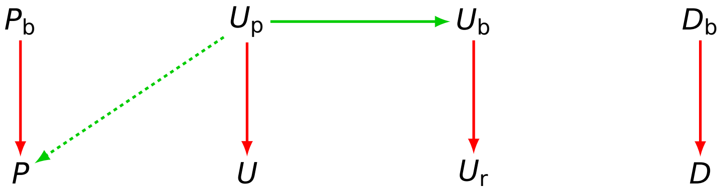
P_b U_b D_b

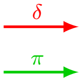
Big Picture

Big Picture

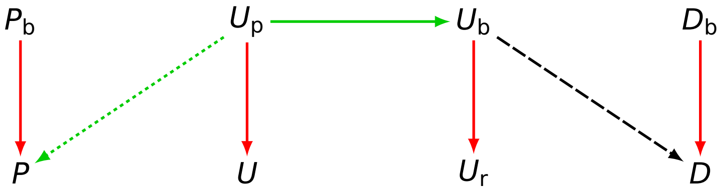


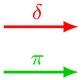
Big Picture



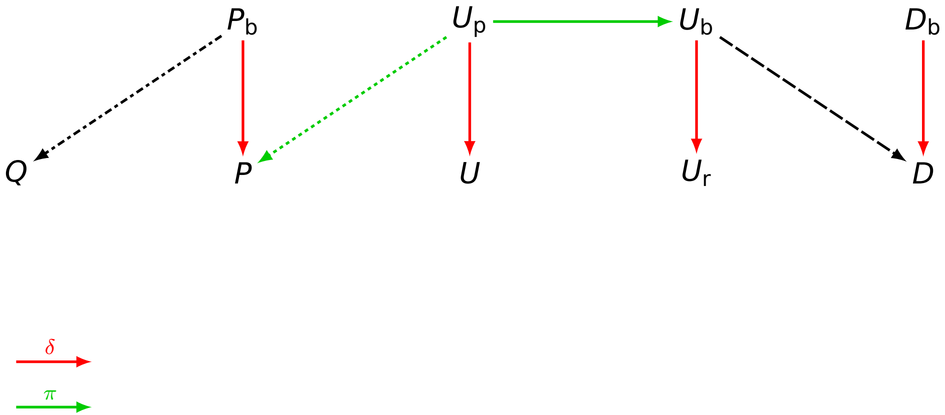


Big Picture

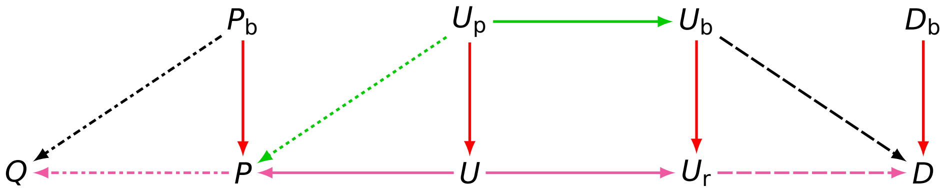


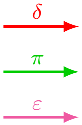


Big Picture

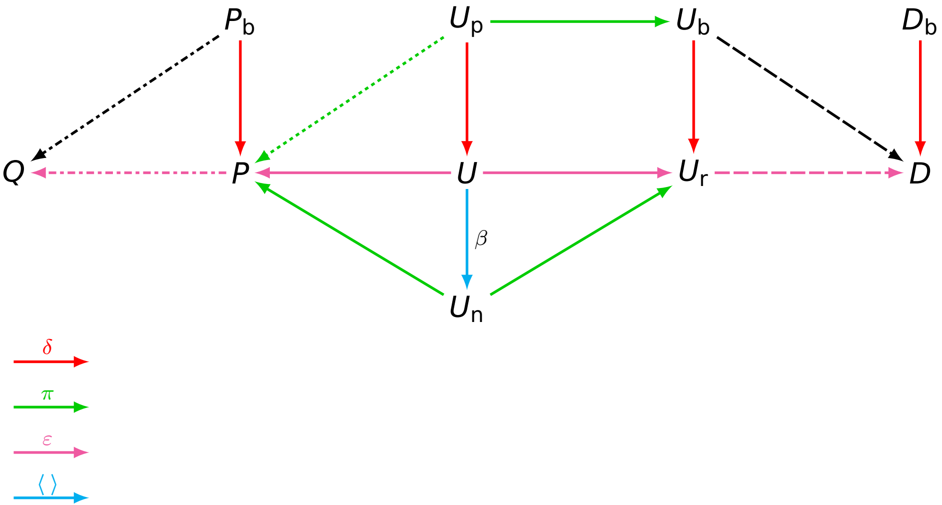


Big Picture

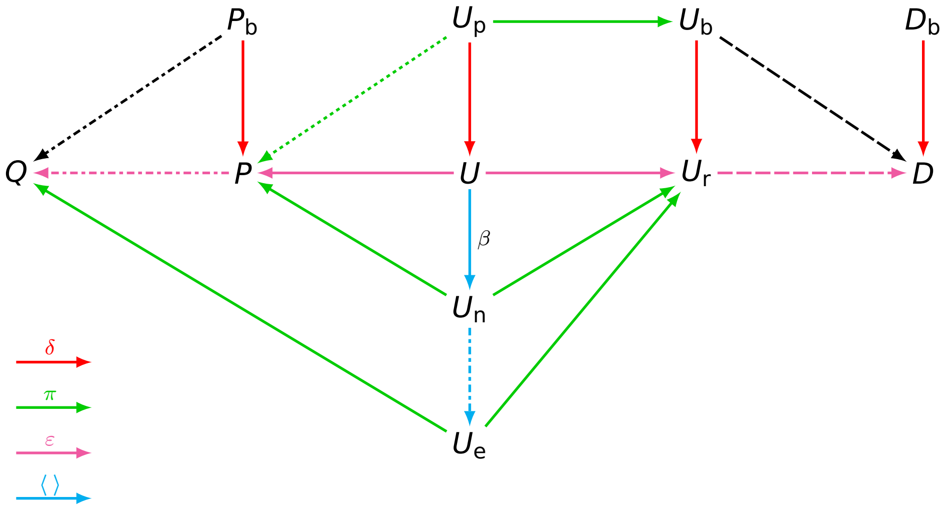




Big Picture

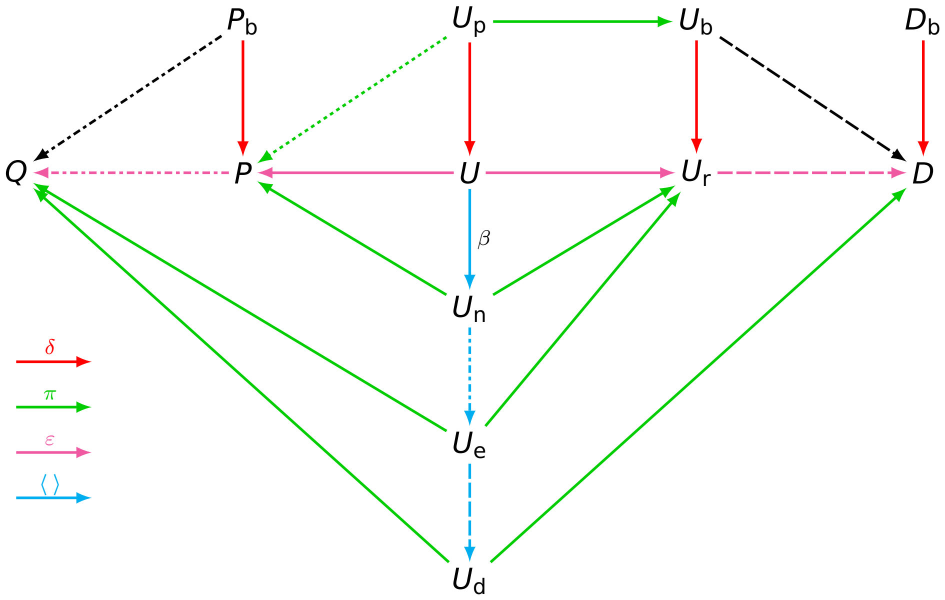


Big Picture



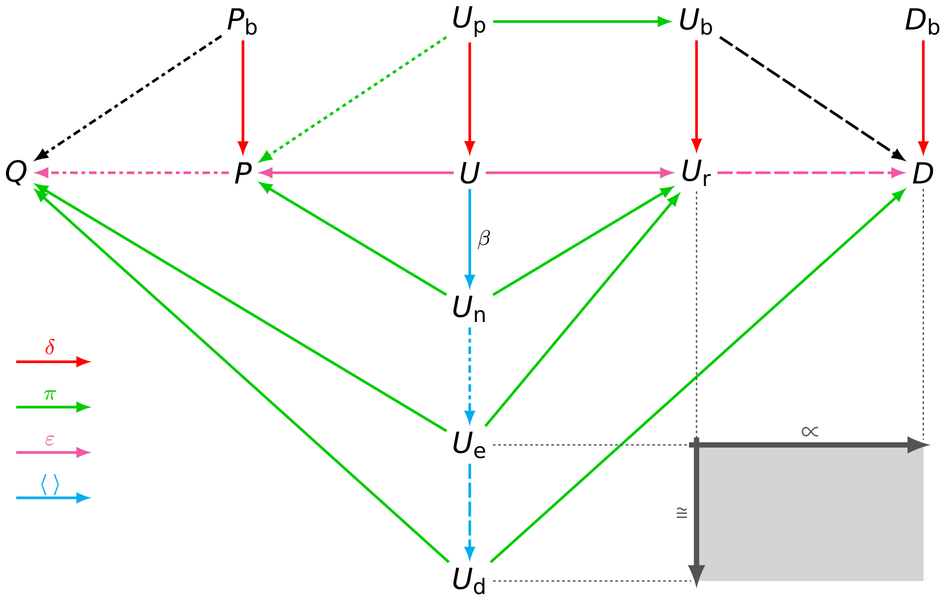
P_b
U_p
U_b
D_b
Q
P
U
U_r
D
β
U_n
U_e
δ
π
ε
$\langle\,\rangle$

Big Picture



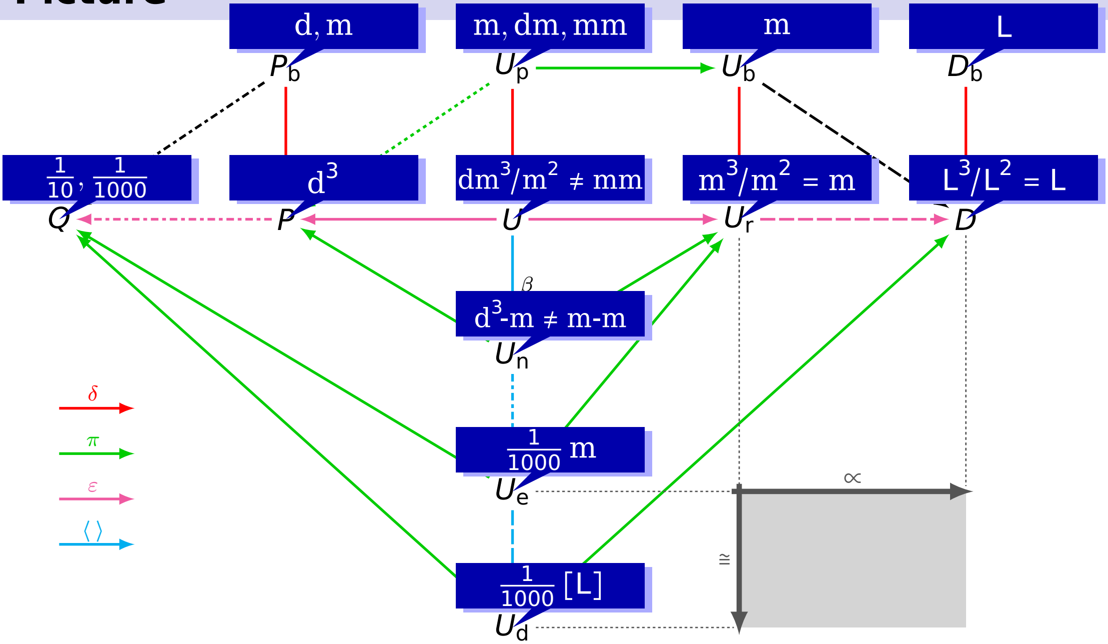
P_b
U_p
U_b
D_b
Q
P
U
U_r
D
β
U_n
U_e
U_d
δ
π
ε
$\langle\,\rangle$

Big Picture



P_b
U_p
U_b
D_b
Q
P
U
U_r
D
β
U_n
U_e
U_d
δ
π
ε
$\langle\,\rangle$
$\propto$
$\cong$

Big Picture


d, m
m, dm, mm
m
L
P_b
U_p
U_b
D_b
$\frac{1}{10}, \frac{1}{1000}$
d^3
$dm^3/m^2 \neq mm$
$m^3/m^2 = m$
$L^3/L^2 = L$
Q
P
U
U_r
D
β
d^3-m ≠ m-m
U_n
$\frac{1}{1000}$ m
U_e
$\frac{1}{1000}$ [L]
U_d
δ
π
ε
$\langle\,\rangle$
∝
≅

Agenda

1. Introduction
2. Unit Algebra
3. **Conversion Relations**
4. Conclusion

Conversion Relations

- Goal: specify conversion factors in a way that privileges no unit.

Conversion Relations

- Goal: specify conversion factors in a way that privileges no unit.
- A **unit conversion** is a ternary relation obeying two axioms:

$$C \subseteq U \times Q \times U$$

1. CODIMENSIONALITY $\quad (u, r, v) \in C \implies \dim(u) = \dim(v)$
2. FUNCTIONALITY $\quad (u, r, v), (u, r', v) \in C \implies r = r'$

Conversion Relations

- Goal: specify conversion factors in a way that privileges no unit.
- A **unit conversion** is a ternary relation obeying two axioms:

$$C \subseteq U \times Q \times U$$

1. CODIMENSIONALITY $\quad (u, r, v) \in C \implies \dim(u) = \dim(v)$
2. FUNCTIONALITY $\quad (u, r, v), (u, r', v) \in C \implies r = r'$

- Write $u \xrightarrow{r}_C v$ for $(u, r, v) \in C$, alluding to categorial diagrams.

Conversion Relations

- Goal: specify conversion factors in a way that privileges no unit.
- A **unit conversion** is a ternary relation obeying two axioms:

$$C \subseteq U \times Q \times U$$

1. CODIMENSIONALITY $\quad (u, r, v) \in C \implies \dim(u) = \dim(v)$
2. FUNCTIONALITY $\quad (u, r, v), (u, r', v) \in C \implies r = r'$

- Write $u \xrightarrow{r}_C v$ for $(u, r, v) \in C$, alluding to categorial diagrams.
- Every triple denotes a rewriting rule: “one u is r vs”.

$$x\,u = x\,(r\,v) = (xr)\,v$$

Conversion Relations

- Goal: specify conversion factors in a way that privileges no unit.
- A **unit conversion** is a ternary relation obeying two axioms:

$$C \subseteq U \times Q \times U$$

1. CODIMENSIONALITY $\quad (u, r, v) \in C \implies \dim(u) = \dim(v)$
2. FUNCTIONALITY $\quad (u, r, v), (u, r', v) \in C \implies r = r'$

- Write $u \xrightarrow{r}_C v$ for $(u, r, v) \in C$, alluding to categorial diagrams.
- Every triple denotes a rewriting rule: "one u is r vs".

$$x\,u = x\,(r\,v) = (xr)\,v$$

- Convertibility with factor 1 is called **coherence**:

$$u \propto_C v \iff u \xrightarrow{\exists r}_C v \qquad\qquad u \cong_C v \iff u \xrightarrow{1}_C v$$

Conversion Closure

- The **conversion closure** of a relation C of the above type is the smallest relation $C^* \supseteq C$ obeying three axioms:

3. MULTIPLICATION
$$u_1 \xrightarrow{r_1}_{C^*} v_1 \wedge u_2 \xrightarrow{r_2}_{C^*} v_2 \implies u_1 u_2 \xrightarrow{r_1 r_2}_{C^*} v_1 v_2$$

4. INVERSE
$$u \xrightarrow{r}_{C^*} v \implies u^{-1} \xrightarrow{r^{-1}}_{C^*} v^{-1}$$

5. DECOMPOSITION
$$u \xrightarrow{\mathrm{pval}(u)}_{C^*} \mathrm{strip}(u)$$

Conversion Closure

- The **conversion closure** of a relation C of the above type is the smallest relation $C^* \supseteq C$ obeying three axioms:

3. Multiplication
$$u_1 \xrightarrow{r_1}_{C^*} v_1 \land u_2 \xrightarrow{r_2}_{C^*} v_2 \implies u_1 u_2 \xrightarrow{r_1 r_2}_{C^*} v_1 v_2$$
4. Inverse
$$u \xrightarrow{r}_{C^*} v \implies u^{-1} \xrightarrow{r^{-1}}_{C^*} v^{-1}$$
5. Decomposition
$$u \xrightarrow{\mathrm{pval}(u)}_{C^*} \mathrm{strip}(u)$$

- The closure of a unit conversion is not necessarily a unit conversion:
  - Codimensionality is preserved, but contradictory factors $u \xrightarrow{r \neq r'}_{C^*} v$ can arise.
- Conversion closure lifts rewriting rules to compound units.

Special Cases

- A conversion is called **defining**, iff left-hand sides are basic and unique:

$$u \xrightarrow{r}_C v \quad \text{is of the form} \quad \lfloor u_0 \rfloor \xrightarrow{r}_C v$$

$$\lfloor u_0 \rfloor \xrightarrow{r}_C v \wedge \lfloor u_0 \rfloor \xrightarrow{r'}_C v' \implies v = v'$$

Special Cases

- A conversion is called **defining**, iff left-hand sides are basic and unique:

$$u \xrightarrow{r}_C v \quad \text{is of the form} \quad \lfloor u_0 \rfloor \xrightarrow{r}_C v$$

$$\lfloor u_0 \rfloor \xrightarrow{r}_C v \wedge \lfloor u_0 \rfloor \xrightarrow{r'}_C v' \implies v = v'$$

  - A defining conversion induces a **rewriting** operation on U_e.
  - A defining conversion C induces a semantic **dependency** order $>_C$ on U_b:

$$\lfloor u_0 \rfloor \propto_C v \wedge \operatorname{supp}\operatorname{root}(v) \ni v_0 \implies u_0 >_C v_0$$

Special Cases

- A conversion is called **defining**, iff left-hand sides are basic and unique:

$$u \xrightarrow{r}_C v \quad \text{is of the form} \quad \lfloor u_0 \rfloor \xrightarrow{r}_C v$$

$$\lfloor u_0 \rfloor \xrightarrow{r}_C v \wedge \lfloor u_0 \rfloor \xrightarrow{r'}_C v' \implies v = v'$$

  - A defining conversion induces a **rewriting** operation on U_e.
  - A defining conversion C induces a semantic **dependency** order $>_C$ on U_b:

$$\lfloor u_0 \rfloor \propto_C v \wedge \operatorname{supp} \operatorname{root}(v) \ni v_0 \implies u_0 >_C v_0$$

- A conversion is called **well-defining**, iff $>_C$ is well-founded.

Special Cases

- A conversion is called **defining**, iff left-hand sides are basic and unique:

$$u \xrightarrow{r}_C v \quad \text{is of the form} \quad \lfloor u_0 \rfloor \xrightarrow{r}_C v$$

$$\lfloor u_0 \rfloor \xrightarrow{r}_C v \wedge \lfloor u_0 \rfloor \xrightarrow{r'}_C v' \implies v = v'$$

  - A defining conversion induces a **rewriting** operation on U_e.
  - A defining conversion C induces a semantic **dependency** order $>_C$ on U_b:

$$\lfloor u_0 \rfloor \propto_C v \wedge \operatorname{supp}\operatorname{root}(v) \ni v_0 \implies u_0 >_C v_0$$

- A conversion is called **well-defining**, iff $>_C$ is well-founded.
  - Rewriting on U_e terminates after a bounded number of steps.
  - Definitions of SI units (e.g., ISO 80000-1) can be read as well-defining.

Conversion Hierarchy

- A conversion is called . . .

1. **consistent** iff its closure is again a conversion;
2. **closed** iff it is its own closure;
3. **finitely generated** iff it is the closure of a finite conversion;
4. **defined** iff it is the closure of a defining conversion;
5. **well-defined** iff it is the closure of a well-defining conversion;
6. **regular** iff it is the closure of an empty conversion.

Conversion Hierarchy

- A conversion is called . . .

1. **consistent** iff its closure is again a conversion;
2. **closed** iff it is its own closure;
3. **finitely generated** iff it is the closure of a finite conversion;
4. **defined** iff it is the closure of a defining conversion;
5. **well-defined** iff it is the closure of a well-defining conversion;
6. **regular** iff it is the closure of an empty conversion.

- Each property in the conversion hierarchy entails the preceding.

Main Results

- Consistency is explosive; contradiction anywhere $\Longrightarrow$ everywhere.

Main Results

- Consistency is explosive; contradiction anywhere $\implies$ everywhere.

- For closed conversions, convertibility and coherence are group congruences.
- Conversion closures are †-compact categories; closed conversions are thin.

Main Results

- Consistency is explosive; contradiction anywhere $\Longrightarrow$ everywhere.
- For closed conversions, convertibility and coherence are group congruences.
- Conversion closures are †-compact categories; closed conversions are thin.
- For closed conversions, root/numerical equivalence entails convertibility/coherence, resp.
- For regular conversions, the converse holds.

Main Results

- Consistency is explosive; contradiction anywhere $\Longrightarrow$ everywhere.
- For closed conversions, convertibility and coherence are group congruences.
- Conversion closures are †-compact categories; closed conversions are thin.
- For closed conversions, root/numerical equivalence entails convertibility/coherence, resp.
- For regular conversions, the converse holds.
- Every closed conversion is finitely generated.
- Every well-defining conversion is consistent, i.e., has a well-defined closure.

Main Results

- Consistency is explosive; contradiction anywhere $\Longrightarrow$ everywhere.
- For closed conversions, convertibility and coherence are group congruences.
- Conversion closures are †-compact categories; closed conversions are thin.
- For closed conversions, root/numerical equivalence entails convertibility/coherence, resp.
- For regular conversions, the converse holds.
- Every closed conversion is finitely generated.
- Every well-defining conversion is consistent, i.e., has a well-defined closure.
- For closed conversions, convertibility encodes the group word problem.
- For well-defined conversions, convertibility is computable efficiently by rewriting to a normal form.

Agenda

1 Introduction

2 Unit Algebra

3 Conversion Relations

4 **Conclusion**

Save the Mars Mission!

$$\lfloor \mathrm{N} \rfloor \xrightarrow{1} \delta(\mathrm{kg}) \lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

$$\lfloor \mathrm{lbf} \rfloor \xrightarrow{1} \lfloor \mathrm{lb} \rfloor g_{\mathrm{n}}$$

$$\lfloor \mathrm{lb} \rfloor \xrightarrow{a} \lfloor \mathrm{g} \rfloor$$

$$g_{\mathrm{n}} \xrightarrow{b} \lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

$$a = 453.59237$$

$$b = 9.80665$$

$$\lfloor \mathrm{N} \rfloor \xrightarrow{1} \delta(\mathrm{kg})\lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

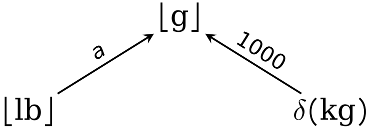
$$\lfloor \mathrm{lbf} \rfloor \xrightarrow{1} \lfloor \mathrm{lb} \rfloor g_{\mathrm{n}}$$

$$\lfloor \mathrm{lb} \rfloor \xrightarrow{a} \lfloor \mathrm{g} \rfloor$$

$$g_{\mathrm{n}} \xrightarrow{b} \lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

$$a = 453.59237$$

$$b = 9.80665$$



Save the Mars Mission!

$$\lfloor \mathrm{N} \rfloor \xrightarrow{1} \delta(\mathrm{kg})\lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

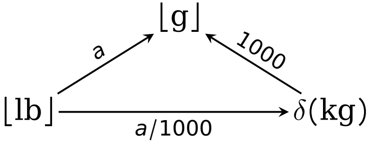
$$\lfloor \mathrm{lbf} \rfloor \xrightarrow{1} \lfloor \mathrm{lb} \rfloor g_{\mathrm{n}}$$

$$\lfloor \mathrm{lb} \rfloor \xrightarrow{a} \lfloor \mathrm{g} \rfloor$$

$$g_{\mathrm{n}} \xrightarrow{b} \lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

$$a = 453.59237$$

$$b = 9.80665$$



Save the Mars Mission!

$$\lfloor \mathrm{N} \rfloor \xrightarrow{1} \delta(\mathrm{kg}) \lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

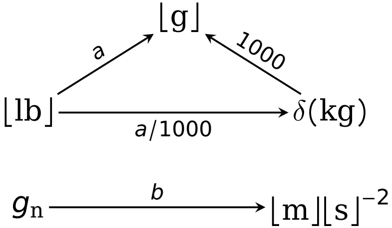
$$\lfloor \mathrm{lbf} \rfloor \xrightarrow{1} \lfloor \mathrm{lb} \rfloor g_{\mathrm{n}}$$

$$\lfloor \mathrm{lb} \rfloor \xrightarrow{a} \lfloor \mathrm{g} \rfloor$$

$$g_{\mathrm{n}} \xrightarrow{b} \lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

$$a = 453.59237$$

$$b = 9.80665$$



Save the Mars Mission!

$$\lfloor \mathrm{N} \rfloor \xrightarrow{1} \delta(\mathrm{kg})\lfloor \mathrm{m}\rfloor\lfloor \mathrm{s}\rfloor^{-2}$$

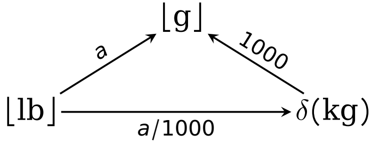
$$\lfloor \mathrm{lbf} \rfloor \xrightarrow{1} \lfloor \mathrm{lb}\rfloor g_{\mathrm{n}}$$

$$\lfloor \mathrm{lb} \rfloor \xrightarrow{a} \lfloor \mathrm{g}\rfloor$$

$$g_{\mathrm{n}} \xrightarrow{b} \lfloor \mathrm{m}\rfloor\lfloor \mathrm{s}\rfloor^{-2}$$

$$a = 453.59237$$

$$b = 9.80665$$



$$g_{\mathrm{n}} \xrightarrow{b} \lfloor \mathrm{m}\rfloor\lfloor \mathrm{s}\rfloor^{-2}$$

$$\lfloor \mathrm{lb}\rfloor g_{\mathrm{n}} \xrightarrow{ab/1000} \delta(\mathrm{kg})\lfloor \mathrm{m}\rfloor\lfloor \mathrm{s}\rfloor^{-2}$$

Save the Mars Mission!

$$\lfloor\mathrm{N}\rfloor \xrightarrow{1} \delta(\mathrm{kg})\lfloor\mathrm{m}\rfloor\lfloor\mathrm{s}\rfloor^{-2}$$

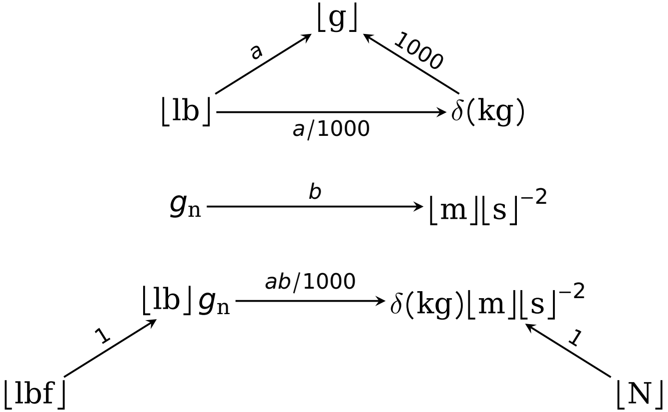
$$\lfloor\mathrm{lbf}\rfloor \xrightarrow{1} \lfloor\mathrm{lb}\rfloor g_{\mathrm{n}}$$

$$\lfloor\mathrm{lb}\rfloor \xrightarrow{a} \lfloor\mathrm{g}\rfloor$$

$$g_{\mathrm{n}} \xrightarrow{b} \lfloor\mathrm{m}\rfloor\lfloor\mathrm{s}\rfloor^{-2}$$

$$a = 453.59237$$

$$b = 9.80665$$



Save the Mars Mission!

$$\lfloor \mathrm{N} \rfloor \xrightarrow{1} \delta(\mathrm{kg}) \lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

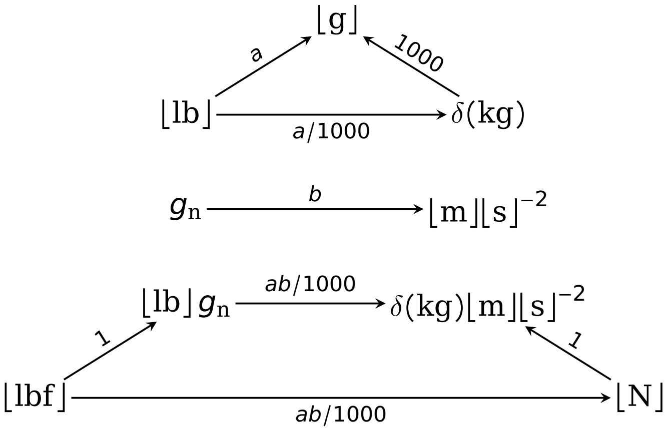
$$\lfloor \mathrm{lbf} \rfloor \xrightarrow{1} \lfloor \mathrm{lb} \rfloor g_{\mathrm{n}}$$

$$\lfloor \mathrm{lb} \rfloor \xrightarrow{a} \lfloor \mathrm{g} \rfloor$$

$$g_{\mathrm{n}} \xrightarrow{b} \lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

$$a = 453.59237$$

$$b = 9.80665$$



Save the Mars Mission!

$$\lfloor \mathrm{N} \rfloor \xrightarrow{1} \delta(\mathrm{kg})\lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

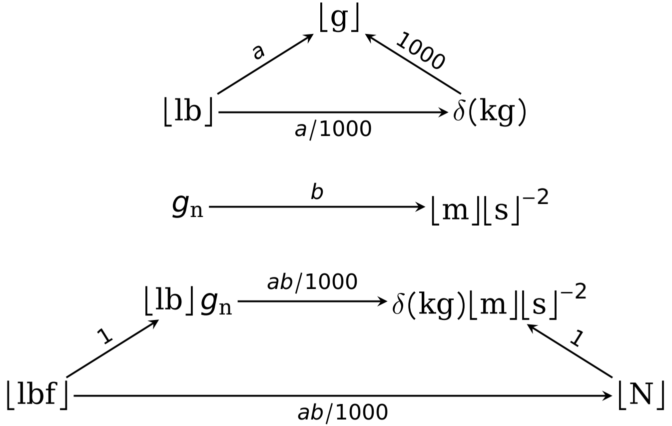
$$\lfloor \mathrm{lbf} \rfloor \xrightarrow{1} \lfloor \mathrm{lb} \rfloor g_{\mathrm{n}}$$

$$\lfloor \mathrm{lb} \rfloor \xrightarrow{a} \lfloor \mathrm{g} \rfloor$$

$$g_{\mathrm{n}} \xrightarrow{b} \lfloor \mathrm{m} \rfloor \lfloor \mathrm{s} \rfloor^{-2}$$

$$a = 453.59237$$

$$b = 9.80665$$



$$ab/1000 = 4.4482216152605$$

Example Symptoms, Revisited

1. ISO 80000-1 states that “1 is not a dimension”, but recognizes “1 as a derived unit”.
   - 1 is a dimension in the same way as $\varnothing$ is a set.
   - 1 is a derived unit in the same way as 0 is a derived natural number.

Example Symptoms, Revisited

1. ISO 80000-1 states that “1 is not a dimension”, but recognizes “1 as a derived unit”.
   - 1 is a dimension in the same way as $\varnothing$ is a set.
   - 1 is a derived unit in the same way as 0 is a derived natural number.
2. ISO 80000-1 defines derived units by equations, but equals cannot be substituted for equals in compound unit expressions.
   - Units normalize naturally; normalized units are monadic.
   - Reductionistic defining equations remain useful for efficient conversion.

Example Symptoms, Revisited

1. ISO 80000-1 states that “1 is not a dimension”, but recognizes “1 as a derived unit”.
   - 1 is a dimension in the same way as $\varnothing$ is a set.
   - 1 is a derived unit in the same way as 0 is a derived natural number.
2. ISO 80000-1 defines derived units by equations, but equals cannot be substituted for equals in compound unit expressions.
   - Units normalize naturally; normalized units are monadic.
   - Reductionistic defining equations remain useful for efficient conversion.
3. ISO 80000-1 defines rad and sr, but treats them as different from each other and 1.
   - *Plane angle* and *solid angle* as quantities are distinct, but not algebraic.
   - The semantic properties of the resp. units are algebraic, but not distinct.

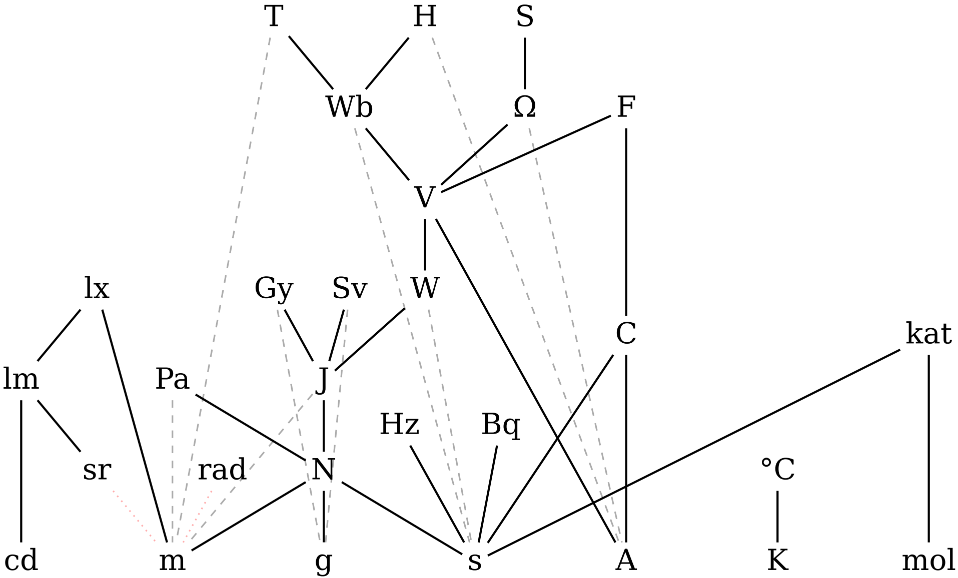
Example Symptoms, Revisited

1. ISO 80000-1 states that "1 is not a dimension", but recognizes "1 as a derived unit".
   - 1 is a dimension in the same way as $\varnothing$ is a set.
   - 1 is a derived unit in the same way as 0 is a derived natural number.
2. ISO 80000-1 defines derived units by equations, but equals cannot be substituted for equals in compound unit expressions.
   - Units normalize naturally; normalized units are monadic.
   - Reductionistic defining equations remain useful for efficient conversion.
3. ISO 80000-1 defines rad and sr, but treats them as different from each other and 1.
   - *Plane angle* and *solid angle* as quantities are distinct, but not algebraic.
   - The semantic properties of the resp. units are algebraic, but not distinct.
4. The SI prefix families are geometric sequences, but cannot be written as powers.
   - Understand them as base prefixes; generate a (free abelian) group.
   - Fun fact: Double prefixes are being considered beyond *-quetta*.

Example Symptoms, Revisited

1. ISO 80000-1 states that “1 is not a dimension”, but recognizes “1 as a derived unit”.
   - 1 is a dimension in the same way as ∅ is a set.
   - 1 is a derived unit in the same way as 0 is a derived natural number.
2. ISO 80000-1 defines derived units by equations, but equals cannot be substituted for equals in compound unit expressions.
   - Units normalize naturally; normalized units are monadic.
   - Reductionistic defining equations remain useful for efficient conversion.
3. ISO 80000-1 defines rad and sr, but treats them as different from each other and 1.
   - *Plane angle* and *solid angle* as quantities are distinct, but not algebraic.
   - The semantic properties of the resp. units are algebraic, but not distinct.
4. The SI prefix families are geometric sequences, but cannot be written as powers.
   - Understand them as base prefixes; generate a (free abelian) group.
   - Fun fact: Double prefixes are being considered beyond *-quetta*.
5. Tools tend to specify conversion by way of one canonical unit per dimension.
   - Conversion relations are (transitively) closed, yet allow multiple disconnected components per dimension.

Bonus Track: The 29 Named SI Units



References

[All+04] E. Allen et al. "Object-Oriented Units of Measurement". In: *SIGPLAN Not.* 39.10 (2004), pp. 384–403. DOI: 10.1145/1035292.1029008.

[Ken96] A. Kennedy. "Programming Languages and Dimensions". PhD Diss. University of Cambridge, 1996. URL: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-391.pdf.

[MBBS20] S. McKeever, O. Bennich-Björkman, and O.-A. Salah. "Unit of measurement libraries, their popularity and suitability". In: *Software: Practice and Experience* 51 (4 2020), pp. 711–734. DOI: 10.1002/spe.2926.

[ISO09] *Quantities and units — Part 1: General*. Standard ISO/IEC 80000-1:2009. International Organization for Standardization, 2009. URL: https://www.iso.org/obp/ui/#iso:std:iso:80000:-1:ed-1:v1:en.

[CGPM16.5] "Resolution 5". In: *Proc. 16th CGPM*. Bureau International des Poids et Mésures, 1979. ISBN: 92-822-2059-1. URL: https://www.bipm.org/en/committees/cg/cgpm/16-1979/resolution-5.

[Ste+99] A. G. Stephenson et al. *Mars Climate Orbiter Mishap Investigation Board Phase I Report*. NASA. 1999. URL: https://llis.nasa.gov/llis_lib/pdf/1009464main1_0641-mr.pdf.

[TL22] B. Trancón y Widemann and M. Lepper. *Towards a Theory of Conversion Relations for Prefixed Units of Measure. Extended report*. 2022. arXiv: 2212.11580.