

Semantic Search for Quantity Expressions

Tom Wiesing

Supervisor: Michael Kohlhase

Co-supervisor: Tobias Preusser

May 20, 2015

110392 Guided Research Applied and Computational
Mathematics & Thesis

Semantic Search for Quantity Expressions

- ▶ Motivation: Problem and State Of The Art

Semantic Search for Quantity Expressions

- ▶ Motivation: Problem and State Of The Art
- ▶ Our Approach: Structure Of The Search Engine

Semantic Search for Quantity Expressions

- ▶ Motivation: Problem and State Of The Art
- ▶ Our Approach: Structure Of The Search Engine
 - ▶ The Unit System

Semantic Search for Quantity Expressions

- ▶ Motivation: Problem and State Of The Art
- ▶ Our Approach: Structure Of The Search Engine
 - ▶ The Unit System
 - ▶ The Search Algorithm

Semantic Search for Quantity Expressions

- ▶ Motivation: Problem and State Of The Art
- ▶ Our Approach: Structure Of The Search Engine
 - ▶ The Unit System
 - ▶ The Search Algorithm
- ▶ The Implementation

Semantic Search for Quantity Expressions

- ▶ Motivation: Problem and State Of The Art
- ▶ Our Approach: Structure Of The Search Engine
 - ▶ The Unit System
 - ▶ The Search Algorithm
- ▶ The Implementation
- ▶ Time for Questions

Motivation (1)

- ▶ We use units every day

Motivation (1)

- ▶ We use units every day
- ▶ We encounter them everywhere:


Motivation (1)

- ▶ We use units every day
- ▶ We encounter them everywhere:
 - ▶ When driving, there are speed limits, for example:




$\frac{\text{km}}{\text{h}}$


Motivation (1)

- ▶ We use units every day
- ▶ We encounter them everywhere:
 - ▶ When driving, there are speed limits, for example:  $\frac{\text{km}}{\text{h}}$
 - ▶ When baking, it often says in recipes something like: “add 3 tea spoons of sugar”


Motivation (1)

- ▶ We use units every day
- ▶ We encounter them everywhere:
 - ▶ When driving, there are speed limits, for example:  $\frac{\text{km}}{\text{h}}$
 - ▶ When baking, it often says in recipes something like: “add 3 tea spoons of sugar”
 - ▶ When shopping for shoes there are different sizes


Motivation (1)

- ▶ We use units every day
- ▶ We encounter them everywhere:
 - ▶ When driving, there are speed limits, for example:  $\frac{\text{km}}{\text{h}}$
 - ▶ When baking, it often says in recipes something like: “add 3 tea spoons of sugar”
 - ▶ When shopping for shoes there are different sizes
- ▶ In scientific papers they occur a lot

Motivation (1)

- ▶ We use units every day
- ▶ We encounter them everywhere:
 - ▶ When driving, there are speed limits, for example:  $\frac{\text{km}}{\text{h}}$
 - ▶ When baking, it often says in recipes something like: “add 3 tea spoons of sugar”
 - ▶ When shopping for shoes there are different sizes
- ▶ In scientific papers they occur a lot
- ▶ everything which somehow models a real system has at least one quantity expression

Motivation (1)

- ▶ We use units every day
- ▶ We encounter them everywhere:
 - ▶ When driving, there are speed limits, for example:  $\frac{\text{km}}{\text{h}}$
 - ▶ When baking, it often says in recipes something like: “add 3 tea spoons of sugar”
 - ▶ When shopping for shoes there are different sizes
- ▶ In scientific papers they occur a lot
- ▶ everything which somehow models a real system has at least one quantity expression
- ▶ everything is quantified

Motivation (2)

- ▶ within one paper, commonly only one type of units is used

Motivation (2)

- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity

Motivation (2)

- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity
- ▶ Just for lengths:

Motivation (2)

- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity
- ▶ Just for lengths:

Motivation (2)

- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity
- ▶ Just for lengths: *Meter*,

Motivation (2)

- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity
- ▶ Just for lengths: *Meter, Inch,*

Motivation (2)

- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity
- ▶ Just for lengths: *Meter, Inch, Foot,*

Motivation (2)

- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity
- ▶ Just for lengths: *Meter, Inch, Foot, Mile,*

Motivation (2)

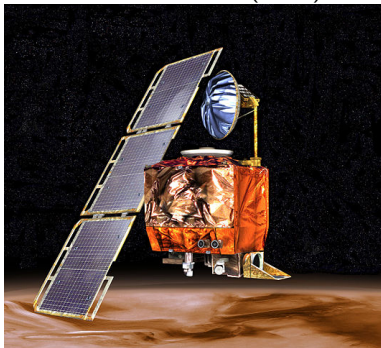
- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity
- ▶ Just for lengths: *Meter, Inch, Foot, Mile, Nautical Mile,*

Motivation (2)

- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity
- ▶ Just for lengths: *Meter, Inch, Foot, Mile, Nautical Mile, ...*
- ▶ This can cause problems when not converting properly

Motivation (2)

- ▶ within one paper, commonly only one type of units is used
- ▶ In general there are **a lot** of **different** units to describe **the same** quantity
- ▶ Just for lengths: *Meter, Inch, Foot, Mile, Nautical Mile, ...*
- ▶ This can cause problems when not converting properly
 - ▶ Mars Climate Orbiter (1999)



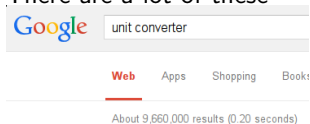
Motivation (3)

Motivation (3)

- ▶ Most common solution: Unit Converters

Motivation (3)

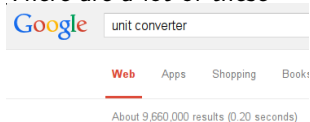
- ▶ Most common solution: Unit Converters
 - ▶ There are a lot of these



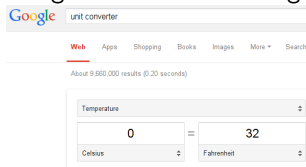
Motivation (3)

- ▶ Most common solution: Unit Converters

- ▶ There are a lot of these



- ▶ Google itself has one integrated



Motivation (4)

- ▶ A lot of user interaction:

Motivation (4)

- ▶ A lot of user interaction:
 - ▶ Problem identification

Motivation (4)

- ▶ A lot of user interaction:
 - ▶ Problem identification
 - ▶ input units & output units

Motivation (4)

- ▶ A lot of user interaction:
 - ▶ Problem identification
 - ▶ input units & output units
 - ▶ not integrated into search process

Motivation (4)

- ▶ A lot of user interaction:
 - ▶ Problem identification
 - ▶ input units & output units
 - ▶ not integrated into search process
- ▶ Wouldn't it be nice:

Motivation (4)

- ▶ A lot of user interaction:
 - ▶ Problem identification
 - ▶ input units & output units
 - ▶ not integrated into search process
- ▶ Wouldn't it be nice:
 - ▶ when searching for $90 \frac{\text{km}}{\text{h}}$

Motivation (4)

- ▶ A lot of user interaction:
 - ▶ Problem identification
 - ▶ input units & output units
 - ▶ not integrated into search process
- ▶ Wouldn't it be nice:
 - ▶ when searching for $90 \frac{\text{km}}{\text{h}}$
 - ▶ we also find $25 \frac{\text{m}}{\text{s}}$

Motivation (4)

- ▶ A lot of user interaction:
 - ▶ Problem identification
 - ▶ input units & output units
 - ▶ not integrated into search process
- ▶ Wouldn't it be nice:
 - ▶ when searching for $90 \frac{\text{km}}{\text{h}}$
 - ▶ we also find $25 \frac{\text{m}}{\text{s}}$
- ▶ This is the kind of search engine we have built

Our Approach (1)

- ▶ What components do we need for a semantic search engine?

Our Approach (1)

- ▶ What components do we need for a semantic search engine?
 1. A *Unit System* that is aware of the different representations of a QE

Our Approach (1)

- ▶ What components do we need for a semantic search engine?
 1. A *Unit System* that is aware of the different representations of a QE
 2. A *Spotter* that finds representations of QEs inside documents

Our Approach (1)

- ▶ What components do we need for a semantic search engine?
 1. A *Unit System* that is aware of the different representations of a QE
 2. A *Spotter* that finds representations of QEs inside documents
 3. A *Search Algorithm* that given a QE finds all its representations in the system

Our Approach (1)

- ▶ What components do we need for a semantic search engine?
 1. A *Unit System* that is aware of the different representations of a QE
 2. A *Spotter* that finds representations of QEs inside documents
 3. A *Search Algorithm* that given a QE finds all its representations in the system
 4. A *Frontend* that allows queries to be made

Our Approach (1)

- ▶ What components do we need for a semantic search engine?
 1. A *Unit System* that is aware of the different representations of a QE
 2. A *Spotter* that finds representations of QEs inside documents
 3. A *Search Algorithm* that given a QE finds all its representations in the system
 4. A *Frontend* that allows queries to be made
- ▶ Spotter is done by *Stiv Sherko*

Our Approach (2)

- ▶ Meta-mathematical model: used to describe structure of mathematics

Our Approach (2)

- ▶ Meta-mathematical model: used to describe structure of mathematics
- ▶ *Theory* = List of *Definitions* (= Constants)

Our Approach (2)

- ▶ Meta-mathematical model: used to describe structure of mathematics
- ▶ *Theory* = List of *Definitions* (= Constants)
- ▶ *Term* = Expression written using definitions from a Theory

Our Approach (2)

- ▶ Meta-mathematical model: used to describe structure of mathematics
- ▶ *Theory* = List of *Definitions* (= Constants)
- ▶ *Term* = Expression written using definitions from a Theory
- ▶ Theories can be related in 2 ways:

Our Approach (2)

- ▶ Meta-mathematical model: used to describe structure of mathematics
- ▶ *Theory* = List of *Definitions* (= Constants)
- ▶ *Term* = Expression written using definitions from a Theory
- ▶ Theories can be related in 2 ways:
 - ▶ *Imports* make Constants from one theory available in another

Our Approach (2)

- ▶ Meta-mathematical model: used to describe structure of mathematics
- ▶ *Theory* = List of *Definitions* (= Constants)
- ▶ *Term* = Expression written using definitions from a Theory
- ▶ Theories can be related in 2 ways:
 - ▶ *Imports* make Constants from one theory available in another
 - ▶ *View* = truth-preserving mapping between theories

Our Approach (2)

- ▶ Meta-mathematical model: used to describe structure of mathematics
- ▶ *Theory* = List of *Definitions* (= Constants)
- ▶ *Term* = Expression written using definitions from a Theory
- ▶ Theories can be related in 2 ways:
 - ▶ *Imports* make Constants from one theory available in another
 - ▶ *View* = truth-preserving mapping between theories
- ▶ Can be displayed in a *Theory Graph*

Our Approach (2)

- ▶ Meta-mathematical model: used to describe structure of mathematics
- ▶ *Theory* = List of *Definitions* (= Constants)
- ▶ *Term* = Expression written using definitions from a Theory
- ▶ Theories can be related in 2 ways:
 - ▶ *Imports* make Constants from one theory available in another
 - ▶ *View* = truth-preserving mapping between theories
- ▶ Can be displayed in a *Theory Graph*
- ▶ *MMT* = software that implements these concepts

Our Approach (2)

- ▶ Meta-mathematical model: used to describe structure of mathematics
- ▶ *Theory* = List of *Definitions* (= Constants)
- ▶ *Term* = Expression written using definitions from a Theory
- ▶ Theories can be related in 2 ways:
 - ▶ *Imports* make Constants from one theory available in another
 - ▶ *View* = truth-preserving mapping between theories
- ▶ Can be displayed in a *Theory Graph*
- ▶ *MMT* = software that implements these concepts
 - ▶ easy to write down theories without programming knowledge

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length
 - ▶ mass

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length
 - ▶ mass
 - ▶ time

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length
 - ▶ mass
 - ▶ time
 - ▶ electric current

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length
 - ▶ mass
 - ▶ time
 - ▶ electric current
 - ▶ temperature

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length
 - ▶ mass
 - ▶ time
 - ▶ electric current
 - ▶ temperature
 - ▶ luminous intensity

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length
 - ▶ mass
 - ▶ time
 - ▶ electric current
 - ▶ temperature
 - ▶ luminous intensity
 - ▶ amount of substance

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length
 - ▶ mass
 - ▶ time
 - ▶ electric current
 - ▶ temperature
 - ▶ luminous intensity
 - ▶ amount of substance
- ▶ but there are also quantities where we just *count*

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length
 - ▶ mass
 - ▶ time
 - ▶ electric current
 - ▶ temperature
 - ▶ luminous intensity
 - ▶ amount of substance
- ▶ but there are also quantities where we just *count*
- ▶ and *dimensionless quantities* (such as Information)

Our Approach: The Unit System (1)

- ▶ Need a *Theory of Quantity Expressions* (QEs)
- ▶ Each quantity has a dimension
- ▶ According to SI there are 7 basic ones:
 - ▶ length
 - ▶ mass
 - ▶ time
 - ▶ electric current
 - ▶ temperature
 - ▶ luminous intensity
 - ▶ amount of substance
- ▶ but there are also quantities where we just *count*
- ▶ and *dimensionless quantities* (such as Information)
- ▶ so we have 9 basic dimensions

Our Approach: The Unit System (2)

- ▶ we can also multiply these to get new dimensions

Our Approach: The Unit System (2)

- ▶ we can also multiply these to get new dimensions
 - ▶ $\text{area} = \text{length} \cdot \text{length}$

Our Approach: The Unit System (2)

- ▶ we can also multiply these to get new dimensions
 - ▶ $\text{area} = \text{length} \cdot \text{length}$
- ▶ similarly we can divide dimensions

Our Approach: The Unit System (2)

- ▶ we can also multiply these to get new dimensions
 - ▶ $\text{area} = \text{length} \cdot \text{length}$
- ▶ similarly we can divide dimensions
 - ▶ $\text{velocity} = \frac{\text{length}}{\text{time}}$

Our Approach: The Unit System (3) - A Theory of Dimensions

Dimension		
dim	:	type
none	:	dim
count	:	dim
length	:	dim
mass	:	dim
time	:	dim
current	:	dim
temperature	:	dim
luminous	:	dim
amount	:	dim
.	:	dim \rightarrow dim \rightarrow dim
/	:	dim \rightarrow dim \rightarrow dim

Our Approach: The Unit System (4)

- ▶ Quantity Expressions can be one of

Our Approach: The Unit System (4)

- ▶ Quantity Expressions can be one of
 1. *primitive unit*, such as Meter

Our Approach: The Unit System (4)

- ▶ Quantity Expressions can be one of
 1. *primitive unit*, such as Meter
 2. *Multiplication* of a (real) number with an existing QE, such as 5 Meter

Our Approach: The Unit System (4)

- ▶ Quantity Expressions can be one of
 1. *primitive unit*, such as Meter
 2. *Multiplication* of a (real) number with an existing QE, such as 5 Meter
 3. *Division* of an existing QE by a (non-zero real) number (equivalent to the above)

Our Approach: The Unit System (4)

- ▶ Quantity Expressions can be one of
 1. *primitive unit*, such as Meter
 2. *Multiplication* of a (real) number with an existing QE, such as 5 Meter
 3. *Division* of an existing QE by a (non-zero real) number (equivalent to the above)
 4. *Product* of two existing QEs such as Newton · Second

Our Approach: The Unit System (4)

- ▶ Quantity Expressions can be one of
 1. *primitive unit*, such as Meter
 2. *Multiplication* of a (real) number with an existing QE, such as 5 Meter
 3. *Division* of an existing QE by a (non-zero real) number (equivalent to the above)
 4. *Product* of two existing QEs such as Newton · Second
 5. *Quotient* of two existing QEs such as $1 \frac{\text{Meter}}{\text{Second}}$

Our Approach: The Unit System (4)

- ▶ Quantity Expressions can be one of
 1. *primitive unit*, such as Meter
 2. *Multiplication* of a (real) number with an existing QE, such as 5 Meter
 3. *Division* of an existing QE by a (non-zero real) number (equivalent to the above)
 4. *Product* of two existing QEs such as Newton · Second
 5. *Quotient* of two existing QEs such as $1 \frac{\text{Meter}}{\text{Second}}$
 6. *Sum* of two existing QEs

Our Approach: The Unit System (5) - A Theory of Quantity Expressions

Quantity Expression	
import Dimension	
QE	: $\text{dim} \rightarrow \text{type}$
QENMul	: $\forall x : \text{dim}. \mathbb{R} \rightarrow \text{QE}(x) \rightarrow \text{QE}(x)$
QENDiv	: $\forall x : \text{dim}. \text{QE}(x) \rightarrow \mathbb{R} \rightarrow \text{QE}(x)$
QEAdd	: $\forall x : \text{dim}. \text{QE}(x) \rightarrow \text{QE}(x) \rightarrow \text{QE}(x)$
QEMul	: $\forall x, y : \text{dim}. \text{QE}(x) \rightarrow \text{QE}(y) \rightarrow \text{QE}(x \cdot y)$
QEDiv	: $\forall x, y : \text{dim}. \text{QE}(x) \rightarrow \text{QE}(y) \rightarrow \text{QE}\left(\frac{x}{y}\right)$

Our Approach: The Unit System (6)

- ▶ we can now easily create theories that define Units, such as a Meter Theory:

Our Approach: The Unit System (6)

- ▶ we can now easily create theories that define Units, such as a Meter Theory:

	Meter
▶	import Quantity Expression
	Meter : QE (length)

Our Approach: The Unit System (6)

- ▶ we can now easily create theories that define Units, such as a Meter Theory:

Meter	
import Quantity Expression	
Meter	: QE (length)

- ▶ we can also define some non-metric lengths:

Our Approach: The Unit System (6)

- ▶ we can now easily create theories that define Units, such as a Meter Theory:

Meter	
import Quantity Expression	
Meter	: QE (length)

- ▶ we can also define some non-metric lengths:

Non SI Lengths	
import Quantity Expression	
Thou : QE (length)	
Foot = QENMul (1000, Thou)	
Yard = QENMul (3, Foot)	
Chain = QENMul (22, Yard)	
Furlong = QENMul (10, Chain)	
Mile = QENMul (8, Furlong)	

Our Approach: The Unit System (7)

- ▶ need to compare units

Our Approach: The Unit System (7)

- ▶ need to compare units
 - ▶ use *Views* (= truth-preserving mappings between theories)

Our Approach: The Unit System (7)

- ▶ need to compare units
 - ▶ use *Views* (= truth-preserving mappings between theories)
 - ▶ For example:

$$\psi = \{ \text{Thou} \mapsto \text{QENMul}(0.0000254, \text{Meter}) \}$$

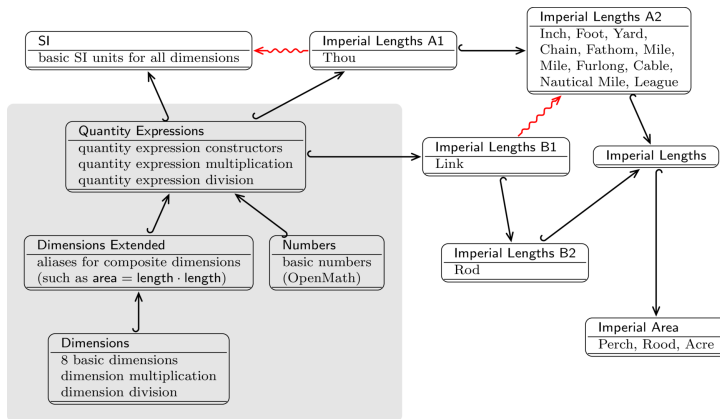
Our Approach: The Unit System (7)

- ▶ need to compare units
 - ▶ use *Views* (= truth-preserving mappings between theories)
 - ▶ For example:

$$\psi = \{ \text{Thou} \mapsto \text{QENMul}(0.0000254, \text{Meter}) \}$$

- ▶ allows conversion

Our Approach: The Unit System (6) - Part of the unit Theory Graph



Our Approach: The Search Algorithm (1)

- ▶ Given:

Our Approach: The Search Algorithm (1)

- ▶ Given:
 - ▶ *Query* (= Quantity Expression) from user

Our Approach: The Search Algorithm (1)

- ▶ Given:
 - ▶ *Query* (= Quantity Expression) from user
 - ▶ *List of QEs* in all documents from spotter

Our Approach: The Search Algorithm (1)

- ▶ Given:
 - ▶ *Query* (= Quantity Expression) from user
 - ▶ *List of QEs* in all documents from spotter
- ▶ *Goal*: Find all QEs equivalent to query

Our Approach: The Search Algorithm (1)

- ▶ Given:
 - ▶ *Query* (= Quantity Expression) from user
 - ▶ *List of QEs* in all documents from spotter
- ▶ *Goal*: Find all QEs equivalent to query
- ▶ *Need*: Efficient way to compare two QEs

Our Approach: The Search Algorithm (1)

- ▶ Given:
 - ▶ *Query* (= Quantity Expression) from user
 - ▶ *List of QEs* in all documents from spotter
- ▶ *Goal*: Find all QEs equivalent to query
- ▶ *Need*: Efficient way to compare two QEs
- ▶ *Idea*: bring QEs to normal form and use efficient indexing

Our Approach: The Search Algorithm (2)

- ▶ Normal form consisting of two components:

Our Approach: The Search Algorithm (2)

- ▶ Normal form consisting of two components:
 - ▶ *scalar* component

Our Approach: The Search Algorithm (2)

- ▶ Normal form consisting of two components:
 - ▶ *scalar* component
 - ▶ (scalar-free) *unit* component in standard units (here: SI)

Our Approach: The Search Algorithm (2)

- ▶ Normal form consisting of two components:
 - ▶ *scalar* component
 - ▶ (scalar-free) *unit* component in standard units (here: SI)
- ▶ use a two-step normalisation process

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain = 10 (22 Yard)

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain = 10 (22 Yard) = ... =
10 (22 (3 (12 (1000 (0.0000254 Meter))))))
- ▶ Fortnight = (2 (7 (24 (60 (60 Second))))))

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain = 10 (22 Yard) = ... =
10 (22 (3 (12 (1000 (0.0000254 Meter))))))
- ▶ Fortnight = (2 (7 (24 (60 (60 Second)))))
- ▶ Substitute this back into the original expression

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain = 10 (22 Yard) = ... =
10 (22 (3 (12 (1000 (0.0000254 Meter))))))
- ▶ Fortnight = (2 (7 (24 (60 (60 Second)))))
- ▶ Substitute this back into the original expression
- ▶ $42 \frac{10(22(3(1000(0.0000254 \text{ Meter}))))}{2(7(24(60(60 \text{ Second}))))}$

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain = 10 (22 Yard) = ... =
10 (22 (3 (12 (1000 (0.0000254 Meter))))))
- ▶ Fortnight = (2 (7 (24 (60 (60 Second))))))
- ▶ Substitute this back into the original expression
- ▶ $42 \frac{10(22(3(1000(0.0000254 \text{ Meter}))))}{2(7(24(60(60 \text{ Second}))))}$
- ▶ Then extract the *scalar* component and compute it:

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain = 10 (22 Yard) = ... =
 $10 (22 (3 (12 (1000 (0.0000254 \text{ Meter}))))))$
- ▶ Fortnight = (2 (7 (24 (60 (60 Second)))))
- ▶ Substitute this back into the original expression
- ▶ $42 \frac{10(22(3(1000(0.0000254 \text{ Meter}))))}{2(7(24(60(60 \text{ Second}))))}$
- ▶ Then extract the *scalar* component and compute it:
- ▶ $42 \frac{10(22(3(12(1000(0.0000254))))))}{2(7(24(60(60))))} = 0.006985$

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain = 10 (22 Yard) = ... =
10 (22 (3 (12 (1000 (0.0000254 Meter))))))
- ▶ Fortnight = (2 (7 (24 (60 (60 Second))))))
- ▶ Substitute this back into the original expression
- ▶ $42 \frac{10(22(3(1000(0.0000254 \text{ Meter}))))}{2(7(24(60(60 \text{ Second}))))}$
- ▶ Then extract the *scalar* component and compute it:
- ▶ $42 \frac{10(22(3(12(1000(0.0000254))))))}{2(7(24(60(60))))} = 0.006985$
- ▶ Continue by extracting the *unit* component: $\frac{\text{Meter}}{\text{Second}}$

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain = 10 (22 Yard) = ... =
10 (22 (3 (12 (1000 (0.0000254 Meter))))))
- ▶ Fortnight = (2 (7 (24 (60 (60 Second)))))
- ▶ Substitute this back into the original expression
- ▶ $42 \frac{10(22(3(1000(0.0000254 \text{ Meter}))))}{2(7(24(60(60 \text{ Second}))))}$
- ▶ Then extract the *scalar* component and compute it:
- ▶ $42 \frac{10(22(3(12(1000(0.0000254))))))}{2(7(24(60(60))))} = 0.006985$
- ▶ Continue by extracting the *unit* component: $\frac{\text{Meter}}{\text{Second}}$
- ▶ Finally multiply these components to get the standard form:

Our Approach: The Search Algorithm (3)

- ▶ Example: Normalise $42 \frac{\text{Furlong}}{\text{Fortnight}}$
- ▶ First, turn all units into standard units by finding an appropriate path in the *theory graph* of units
- ▶ Furlong = 10 Chain = 10 (22 Yard) = ... =
10 (22 (3 (12 (1000 (0.0000254 Meter))))))
- ▶ Fortnight = (2 (7 (24 (60 (60 Second)))))
- ▶ Substitute this back into the original expression
- ▶ $42 \frac{10(22(3(1000(0.0000254 \text{ Meter}))))}{2(7(24(60(60 \text{ Second}))))}$
- ▶ Then extract the *scalar* component and compute it:
- ▶ $42 \frac{10(22(3(12(1000(0.0000254))))))}{2(7(24(60(60))))} = 0.006985$
- ▶ Continue by extracting the *unit* component: $\frac{\text{Meter}}{\text{Second}}$
- ▶ Finally multiply these components to get the standard form:
- ▶ $0.006985 \frac{\text{Meter}}{\text{Second}}$

Our Approach: The Search Algorithm (4)

- ▶ compare the spotted QEs and the query in normal form

Our Approach: The Search Algorithm (4)

- ▶ compare the spotted QEs and the query in normal form
- ▶ To save time:

Our Approach: The Search Algorithm (4)

- ▶ compare the spotted QEs and the query in normal form
- ▶ To save time:
 - ▶ *cache* the normal form of each unit (only find paths once)

Our Approach: The Search Algorithm (4)

- ▶ compare the spotted QEs and the query in normal form
- ▶ To save time:
 - ▶ *cache* the normal form of each unit (only find paths once)
 - ▶ *cache* the normal form of the spotted QEs

Our Approach: The Search Algorithm (4)

- ▶ compare the spotted QEs and the query in normal form
- ▶ To save time:
 - ▶ *cache* the normal form of each unit (only find paths once)
 - ▶ *cache* the normal form of the spotted QEs
- ▶ We normalise to SI units here, but we can freely choose

The Implementation

- ▶ Components of the search process

The Implementation

- ▶ Components of the search process
 - ▶ Frontend (in the browser)

The Implementation

- ▶ Components of the search process
 - ▶ Frontend (in the browser)
 - ▶ when queried, this sends a QE to the backend

The Implementation

- ▶ Components of the search process
 - ▶ Frontend (in the browser)
 - ▶ when queried, this sends a QE to the backend
 - ▶ Backend (in scala) uses MMT to normalise query

The Implementation

- ▶ Components of the search process
 - ▶ Frontend (in the browser)
 - ▶ when queried, this sends a QE to the backend
 - ▶ Backend (in scala) uses MMT to normalise query
 - ▶ Searches the harvests (provided by Stiv's Spotter)

The Implementation

- ▶ Components of the search process
 - ▶ Frontend (in the browser)
 - ▶ when queried, this sends a QE to the backend
 - ▶ Backend (in scala) uses MMT to normalise query
 - ▶ Searches the harvests (provided by Stiv's Spotter)
 - ▶ Browser displays results

Conclusion: The Implementation (2)

- ▶ Supported units only limited by theory graph

Conclusion: The Implementation (2)

- ▶ Supported units only limited by theory graph
- ▶ Can be easily extended

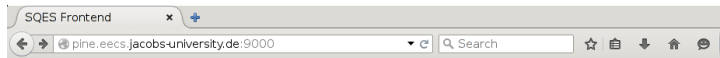
Conclusion: The Implementation (2)

- ▶ Supported units only limited by theory graph
- ▶ Can be easily extended
- ▶ User no longer needs to think about units to find

Conclusion: The Implementation (2)

- ▶ Supported units only limited by theory graph
- ▶ Can be easily extended
- ▶ User no longer needs to think about units to find
- ▶ Demo at
`http://pine.eecs.jacobs-university.de:9000/`

The Implementation (3)



SQES Demo

328.0839895013123

Foot

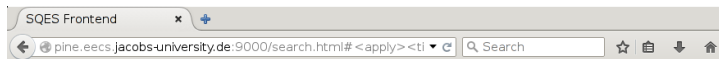
ImperialLengthA2?Foot

Press enter or this

Search

button to search

The Implementation (4)



SQES Demo

<http://arxmliv.kwarc.info/files/0511/astro-ph.0511100/astro-ph.0511100.xhtml#S2.SS3.p1.m5>

<rdf:xml:literal><apply><times></times><cn type="real">100.0</cn><csymbol cd="SIBase">Meter</csymbol></apply></rdf:xml:literal>

<http://arxmliv.kwarc.info/files/0510/physics.0510152/physics.0510152.xhtml#S6.p2.m61>

<rdf:xml:literal><apply><times></times><cn type="real">100.0</cn><csymbol cd="SIBase">Meter</csymbol></apply></rdf:xml:literal>

<http://arxmliv.kwarc.info/files/0510/physics.0510214/physics.0510214.xhtml#S4.p5.m4>

<rdf:xml:literal><apply><times></times><cn type="real">100.0</cn><csymbol cd="SIBase">Meter</csymbol></apply></rdf:xml:literal>

Thank You For Listening!

Image sources:

- ▶ http://www.gettingaroundgermany.info/g_imgs/z274.gif
- ▶ http://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Mars_Climate_Orbiter_2.jpg/528px-Mars_Climate_Orbiter_2.jpg