Adding Apples and Oranges

January 30, 2002

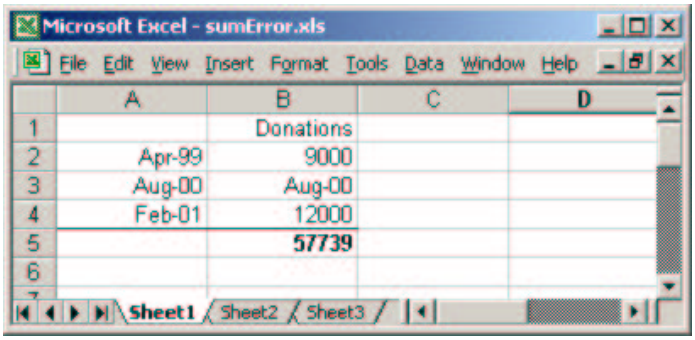
Martin Erwig
Margaret Burnett

Prevent errors in spreadsheets

1

January 30, 2002

An Example



According to Excel:
Aug-00 = 36739

2

The Problem

Two aspects of **spreadsheets**:

- up to 40% contain errors
- 55,000,000 end user programmers in 2005 (in the United States)

⇒ Strong need to make spreadsheets more reliable

The Challenge

- Type Systems can be extremely helpful in error detection/prevention, but:

- An abstract concept of types is very difficult to impart on end users

Type System

End user

The Idea

Use vocabulary from the spreadsheet to communicate with the end user

Example:
Column headers

'units'

Adding 'Specific' Apples and Oranges

Microsoft Excel - harvest.xls

File Edit View Insert Format Tools Data Window Help

	A	B	C	D	E
1		Fruit			
2	Month	Apple	Orange	Total	
3	May	8	11	19	
4	June	20	50	70	
5	Total	28	61	89	
6					

Sheet1 Sheet2 Sheet3

Goal: B3+C3 ✓

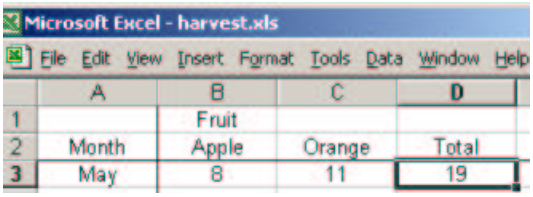
B3+C4 ERROR!

Formalization Roadmap

Goals: (1) associate cells with units
(2) determine unit-correctness

- Initial unit information is given by headers
- Multiple/nested headers and operations lead to complex units
- Define unit normal form and simplification rules
- Notion of unit-correctness: All cells have units in normal form

Units

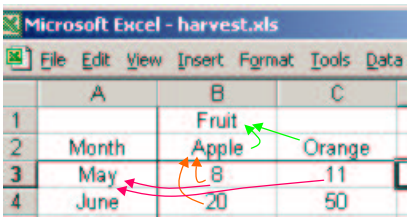


- Values and cells have units
  8:Apple B3::Apple — unit **judgments**
- Units can be nested
  B2::Fruit B3::Fruit[Apple] — **dependent** units
- Cells might have multiple units
  B3::Month[May]&Fruit[Apple] — **and** units
- Operations can generalize units
  B3+C3:Apple|Orange
  B3+C3:Fruit — **or** units

And also: units **1** and **ε**

Headers & Units

- All values in a spreadsheets are units
  Fruit Apple June 8 50 ...
- A **header** is a label for a group of cells and defines unit information
  Fruit: B2 C2
  Apple: B3 B4
  May: B3 C3
- Chains of headers define **dependent units**
  Fruit[Apple]
  Month[May]



- A cells can have multiple headers ⇒ **& units**
  B3::Month[May]&Fruit[Apple]

Well-formed Units

Unrelated units can be combined with **&**

Month[May]&Fruit[Apple]
Month[May]&Fruit
Month&Fruit
Apple&Orange

} well formed

Units of the same nesting level (>1) that have the same ancestors can be combined with |

Fruit[Apple]|Fruit[Orange]
Month[May]|Month[June]

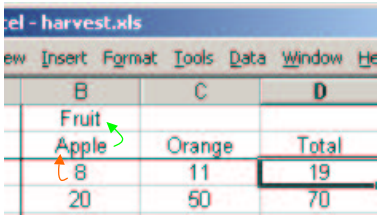
} well formed

Month|Fruit
Fruit[Apple[Green]]|Fruit[Orange]

Unit Inference

(1) A cell without a header has unit 1

B1::1 D2::1



(2) Units propagate through references

(3) A cell with a header that contains value V and has unit U has unit U[V]

B3::Fruit[Apple]

(4) Operations have their own unit transformations

$+_u(U_1,U_2;U) = (U_1|U_2)\&U$

$*_u(U_1,1;U) = U_1\&U$

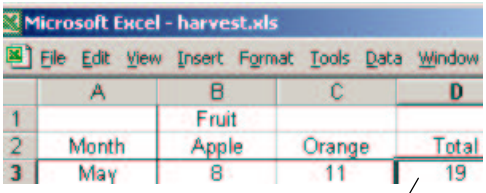
$*_u(1,U_2;U) = U_2\&U$

D2::(Fruit[Apple]|Fruit[Orange])&1

Unit Simplification

$U_1\&U_2 = U_2\&U_1$ $U_1\|U_2 = U_2\|U_1$		commutativity
$(U_1\&U_2)\&U_3 = U_1\&(U_2\&U_3)$ $(U_1\|U_2)\|U_3 = U_1\|(U_2\|U_3)$		associativity
$U\&U = U$ $U\|U = U$		idempotency
$U\&(U_1\|U_2)$	$= (U\&U_1)\|(U\&U_2)$	distributivity
$1\&U$	$= U$	unit
$U[U_1]\|U[U_2]$	$= U[U_1\|U_2]$	factorization
$U[U_1\|\ldots\|U_k]$	$= U$	generalization(*)
$(U_1[U_2])[U_3]$	$= U_1[U_2[U_3]]$	linearization

Example



D3=B3+C3

By header (rule 3):

B3::Month[May]&Fruit[Apple]
C3::Month[May]&Fruit[Orange]

By operation + (rule 4):

D3::Month[May]&Fruit[Apple] |
Month[May]&Fruit[Orange]

Simplification (distrib.,factor.,general.):

D3::Month[May]&(Fruit[Apple]|Fruit[Orange])
D3::Month[May]&(Fruit[Apple|Orange])
D3::Month[May]&Fruit ✓

Two Products

Offline tool to check legacy spreadsheets

possible now ✓

Online unit checking integrated into Microsoft Excel

need permission/
support from Microsoft...

Future Work

- Units are more fine-grained than types
  ⇒ Define unit-aware semantics to obtain appropriate soundness results

- Header inference
  (1) Predefined unit information
  May:Month Blue:Color ...
  (2) Infer headers from table formatting actions
  (3) Spatial analysis
  Infer headers from formula positions

One Conclusion
(from a PADL point of view)

Adding
Functional Programming/Type Systems
and
End-User Programming

✓