A Spreadsheet for Everyday Symbolic Reasoning

Iliano Cervesato
Deductive Solutions
iliano@deductivesolutions.com
The Traditional Spreadsheet

Benefits

- Simple access to complex \textit{numerical} calculations
  - Intuitive interface
  - No formal training needed
  - Gentle learning curve
- Effective decision support for \textit{numerical} data
  - Financial analysis, budgets, grades, inventories, …
- Ubiquitous
  - Over 50M users
  - Only recently surpassed by web browsers and mailers

Opportunities

- Simple access to \textit{symbolic} calculations/reasoning
- Effective decision support for \textit{symbolic} data
Objectives of this Work

- Extend the spreadsheet with symbolic reasoning
  - Support symbolic decision-making
  - Provide functionalities to manipulate data symbolically
    - Logical language
    - Operational interpretation
    - Interface commands
  - Same ease of use as traditional spreadsheet
- Seamless integration into current model
  - Not a separate application
Results

- Extension of the traditional spreadsheet with:
  - Expressions over first-class tabular data
    - Datalog with negation, constraints, calculated values, lists
    - Equational relational algebra (extended)
  - Like database, but queries results permanently displayed
  - Efficient evaluation and update propagation
    - Guaranteed termination
  - Explanation facilities
  - Extended user interface
  - Good feedback from preliminary user testing
Rest of this Talk

- What is a spreadsheet?
- Extended core functionalities
  - Relational/Logical expressions
  - Evaluation / Updates / Explanation
- Extended user interface
  - Design methodology
  - Extensions
- User testing
Historical Attempts

- 1982: LogiCalc [Kriwaczek]
  - Spreadsheet in MicroProlog
  - + relational views, integrity constraints, bidirectional variables, symbolic manipulations, complex objects
  - Teletype interface
- 1986: [van Emden]
  - Incremental queries, exploratory programming
- 1989: PERPLEX [Spenke & Beilken]
  - Bidirectional integrity constraints
- Then not much … until now!
What is a Spreadsheet?

Mathematical model for

- Scalar spreadsheet
- Array formulas
- Relational support
Scalar Spreadsheets

A simple functional language without recursion

- 16,777,216 glorified calculators

Functionalities

- Input
  - Cells, Expressions
- Calculate
  - Turn entered expressions into displayed values
- Update
  - Propagate changes
- Explanation (audit)
  - Catch errors
Spreadsheet Model

- Scalar expressions
  - A2 * 9/5 + 32

- Spreadsheet:
  - s : Cell \rightarrow Expr
    - No circular references

- Dependency graph:
  - DG_s
    - Representation of s that highlights cell dependencies
Evaluation

Environment: Env = Cell $\rightarrow$ Val
Evaluation: eval: s $\rightarrow$ Env

- Best performed on dependency graph
  - Fixpoint calculation
    - Starts from undefined environment
    - $\#$ iterations = longest path in DGs
  - Cost = $\mathcal{O}(\text{used\_cells})$
    - Under semi-naive strategy
Updates

- Determine tainted cells
  - Using dep. graph
- Evaluation starting from tainted environment
- Cost = $O(tainted\_cells)$
  - Under semi-naïve strategy
Explaination

Why does A2 show 212?

- Commands to navigate DG_s from given cell
  - Highlight cells on which A2 depends
  - ... and those on which they depend
  - ... and those on which they depend
  - ... and those on which they depend
  - ...

- ...
Array Formulas

- Expressions associated to a block of cells
  - A44 := \( \frac{\text{SUM}(A2:A43)}{42} \)
  - B2:B43 := \( A2:A43 \times \frac{9}{5} + 32 \)

- Map to scalar formulas
  - No circularity at that level
  - Inherit evaluation and update

- Immature user interface
Relational Support

- “Data List” / “Databases” / ...
  - Minimal support for manipulating tabular data
    - Insertion wizard
    - Sorting
    - Selection
    - Import from other applications
  - Second class-objects
    - Functionalities as commands, not operations
  - No functions over multiple tables
    - No join
The Deductive Engine

- First-class relations
  - Relational expressions
  - Integration
- Logical counterpart
  - Datalog without recursion
  - Logical updates
  - Explanation as proof-search
- Deductive spreadsheet
  - Recursion
  - Bounded termination
Relations

- Interpret rows as records, columns as attributes
- Or the other way around
- Nothing new
Relational Expressions

- Associated to cell blocks
  - Like array formulas

- Manipulate relations as a whole
  - Union, difference, projection, selection, join
  - *Show all flights between Delta hubs less than 500 miles apart*
    
    $\pi_{\text{hub1.City}, \text{hub2.City}} \sigma_{\text{directFlight.Distance}<500, \text{hub1.Airline}=\text{"Delta"}, \text{hub2.Airline}=\text{"Delta"}}$

- Non-deterministic ordering
  - No duplicates

- Minor extension for calculated projection attributes
  - Result is treated as a set
Interface to Usual Formulas

- Coercion from (array) formula to relation
  - $<e>$: compute $e$ and interpret it as a relation
- Coercion from relational exp. to (array) formula
  - $[r]$: compute $r$ and interpret it as an array
    - Ordering is non-deterministic
    - Add SORT as a new array operation
- Traditional formulas also in selection/projection attributes
- Relational expressions can appear within formulas
- Formulas can appear within relational expressions
Relational Spreadsheet

\[ s : \text{Partition(Cell)} \rightarrow \text{ArrayExp} \cup \text{RelExp} \]

- Cannot be reduced to scalar spreadsheet
- Several notions of dependency graph
  - Cell level
  - Relation level
    - No circularity
  - Attribute level
Functionalities

- **Evaluation**
  - $\text{Env} = \text{Partition}(\text{Cell}) \rightarrow \text{Val} \cup \text{RelVal}$
  - $\text{Eval} : s \rightarrow \text{Env}$
  - $\text{Cost} = \mathcal{O}(\text{records}^{\text{max}_\text{join}})$
    - Semi-naïve evaluation

- **Update**
  - Identifies added/removed records
  - Start reevaluation from those

- **Explanation**
  - Similar to traditional spreadsheet
  - Inadequate
Logical Interpretation

- Rel. algebra equivalent to recursion-free Datalog
  
  Show all flights between Delta hubs less than 500 miles apart
  
  \[
  \text{shortDeltaFlight(From,To) } \leftarrow \\
  \text{directFlight(From,To,Dist) } \land \\n  \text{Dist < 500 } \land \\n  \text{hub(From, "Delta") } \land \\n  \text{hub(To, "Delta")}
  \]

- Body literals can be negated
  - Stratified Datalog

- Extension with constraints
  - Generic
  - Head: operate on head-only variable

- Variables subject to safety restrictions
So What?

Harness wide array of logical tools

- 40 years of logic programming
- Logical interpretation of
  - Evaluation
    - Logical inference
  - Updates
    - Optimized evaluation
  - Explanation
    - proof-search
Evaluation Revisited

- Logical consequences computed as
  - Fixpoint of functional on logical interpretations
  - Bottom-up evaluation of logic programs

- Terminating
  - Fast strategies
    - Semi-naïve strategy
  - Used in deductive databases

- Scales to
  - Stratified negation
  - Safe constraints
  - Surrounding scalar/array formulas
## Updates Revisited

- Incremental evaluation at heart of semi-naïve strategy
  - Optimization
- Adapts smoothly to generic updates
  - Positive updates
  - Negative updates
Explanation Revisited

- Display argument for computed record
  - Proof search
  - Top-down evaluation of logic programs
- Flexible explanation mechanism
  - Why is this record there?
  - Why isn’t this record there?
  - May contain variables
  - Proof of generic queries
The Deductive Spreadsheet

- Allow recursion
  - Subject to stratification

*Show all pairs of cities connected by air*

\[
\text{indirect}(\text{From}, \text{To}) \leftarrow \text{directFlight}(\text{From}, \text{To}, _). \\
\text{indirect}(\text{From}, \text{To}) \leftarrow \text{directFlight}(\text{From}, \text{Mid}, _) \& \text{indirect}(\text{Mid}, \text{To})
\]

- Strictly more expressive
  - Opens the door to a whole new class of problems
  - Even more so by exploiting spreadsheet environment
    - Overlapping traditional formulas
Examples of Expressiveness

- Any relational expression
  - Any SQL query
- Recursive queries
  - Transitive closure problems
    - Path in a graph
    - Travel planning
    - Hierarchies
    - Course requirements
    - Readiness of troops, …
- Bill of Material problem
- Workflow problem
- Meeting planner
- Anti-trust problem
Extensions

- **Head constraints in recursive clauses**

  *Show distance of trip*

  \[
  \text{indirect}(\text{From}, \text{To}, \text{Dist}) \leftarrow \text{directFlight}(\text{From}, \text{To}, \text{Dist}).
  \]

  \[
  \text{indirect}(\text{From}, \text{To}, \text{Dist}) \leftarrow \text{directFlight}(\text{From}, \text{Mid}, \text{Dist’}) \land \text{indirect}(\text{Mid}, \text{To}, \text{Dist’’}) \land \text{Dist} = \text{Dist’} + \text{Dist’’}
  \]

  - Non-terminating in general
  - Put user-defined bound on recursion for these clauses

- **Flat lists**

  *Show itinerary*

  \[
  \text{indirect}(\text{From}, \text{To}, [\text{From}, \text{To}]) \leftarrow \text{directFlight}(\text{From}, \text{To}, _).
  \]

  \[
  \text{indirect}(\text{From}, \text{To}, [\text{From}, \text{Mid}|\text{Rest}]) \leftarrow \text{directFlight}(\text{From}, \text{Mid}) \land \text{indirect}(\text{Mid}, \text{To}, [\text{Mid}|\text{Rest}])
  \]

  - Treated in the same way

- **Embedded implication**
### The User Interface

- **Design methodology**

- **Initial design**
  - Most modern spreadsheets have nearly identical interfaces
  - Generic deductive extension
    - Demonstrated on Excel 2000
Interface Design Methodology

- Traditional approaches
  - Experts design user interface
    - We are not HCI experts
  - Refined through extensive user testing
    - No time/resources at this stage
- Lightweight approximate methods
  - Meant for application designers
  - Provide vocabulary for concepts and objectives
  - Obtain adequate first-cut
    - Validate/refine later using traditional approaches
Cognitive Dimensions

- “Discussion tools” for cognitive concepts
  - Viscosity
  - Consistency
  - Hard mental operations
  - Hidden dependencies, …

- Vocabulary to make decisions
  - Evaluate cognitive effect
  - Plan trade-offs

- Scales to make rough measurements
Attention Investment Model

- Psycho-economic model to anticipate user behavior
  - Embracing novelty = investment of attentional effort
  - Will do if perceived pay-off > perceived risk

- Pay-off: larger class of solvable problems

- Costs:
  - Shifting to logical/relational mindset
  - Learning new syntax

- Risk: problem still not solvable

- Target audience
  - Needed skills
    - Tabular information, select cell ranges, comfortable with formulas
  - Advanced and intermediate users
Deductive Layout

- Nearly unchanged
  - No cognitive penalty
- Couple of new context-sensitive menu items
  - “Define Relation …”
    - Give names to relation and attributes
    - Insert it in “defined predicates” list
    - Insert captions
  - “Explain”
- Graphical construction of formulas
Textual Language of Formulas

Two alternatives

- Gives flexibility to user
- Embellished Datalog
  - indirect(From, To) \textbf{IF} directFlight(From, To, _).
  - indirect(From, To) \textbf{IF} directFlight(From, Mid, _) AND indirect(Mid, To)
- SQL-like language
  - indirect(To, From) = directFlight \textbf{UNION}
    - \textbf{SELECT} directFlight.From, indirect.To \textbf{FROM} directFlight, indirect
    - \textbf{WHERE} directFlight.To = indirect.From

- Final choice to be guided by user feedback
Entering Formulas

- Typing in the formula bar
  - Syntax check “as-you-type”
    - Visual feedback
  - Autoformat
  - Precise error reporting
- Clicking around
- Wizards
- Cut and paste
Mouse-Assisted Definition

- Construct formula with a few mouse clicks
  - Names from “predicate list” or spreadsheet

- Identify variables by dragging
- Click constraints in

```
=indirect(From,To)
IF directFlight(From,To,_).
=indirect(From,To)
IF directFlight(Var1,Var2,Var3)
```
Wizard-Assisted Definition

- Enter formula in wizard
- Mouse assisted shortcuts available

### Clause Definition

**Head**  indirect(From,To)

**Body**

- **Conjunct 1**: directFlight(From, Mid, _)
- **Conjunct 2**: indirect(Mid, Var5)
- **Conjunct 3**: Mid <> “LAX”
- **Conjunct 4**: 

**Head** is true only if all the **conjuncts** in **Body** are true

**Conjunct 4**: each **conjunct** is a **predicate** or a **constraint**

- Right-click on box for **defined predicates**
- Drag variables to define **constraints**
- Click on $f_x$ for **abbreviated forms**

Define another **clause**
Explanation Facilities

- Invoked using right-click menu
- Displays proof tree
  - Color-coded feedback in spreadsheet
  - Browsable
- Allow entering arbitrary queries
- Allows saving result

Query: indirect("JFK", To)

indirect("JFK", "LAX")
  indirect("SFO", "LAX")
    directFlight("SFO", "LAX")
  directFlight("JFK", "SFO")
    indirect("JFK", "LAX")
    directFlight("JFK", "SFO")

indirect("SFO", "LAX")
  directFlight("SFO", "LAX")

IF directFlight("JFK", "SFO")
AND indirect("SFO", "LAX").
Productivity Tools

Connection graph

Connection Graph

LGA 211 → DCA 540 → ATL 2130
JFK 181 → BWI 2580 → SFO
    2437 → LAX 1657 → MSY
IAD 3850 → CDG 2252
    2580 → VCE 519
    3640 → EWR 2560

Query 1
- directFlight(F,T,D)
- D > 2500

Query 2
- directFlight(F,T,D)
- AND D > 2500

More soon
- Flow graph
- …
Preliminary User Testing

- 8 volunteers
  - 3 advanced
  - 2 intermediate
  - 2 beginners — *NOT in target audience*
  - 1 theoretical computer scientist …

- Outline of experiment
  1. Background questionnaire
  2. Illustration of Deductive Spreadsheet
  3. Walk through example and user interface

Collected feedback at each stage
Feedback

- Advanced users
  - Followed example and suggested applications
  - General approval of user interface
  - Interested in all aspects of the Deductive Spreadsheet
  - Would use the Deductive Spreadsheet if it were available

- Intermediate users
  - Followed example and suggested applications
  - Disapproved of choice of some keywords in interface
  - Interest in many aspects of the Deductive Spreadsheet

- Beginners — *NOT in target audience*
  - Appreciated general objectives but difficulties with example
  - Trouble with wording of interface
  - Lot of interest in basic relational inference
    - Demanded simpler interface
Future Work

- Prototype
- Enhancements to User Interface
- Experimental assessment
  - User testing
  - Performance
  - Problem base
- Integration of other notions of “deductive”