

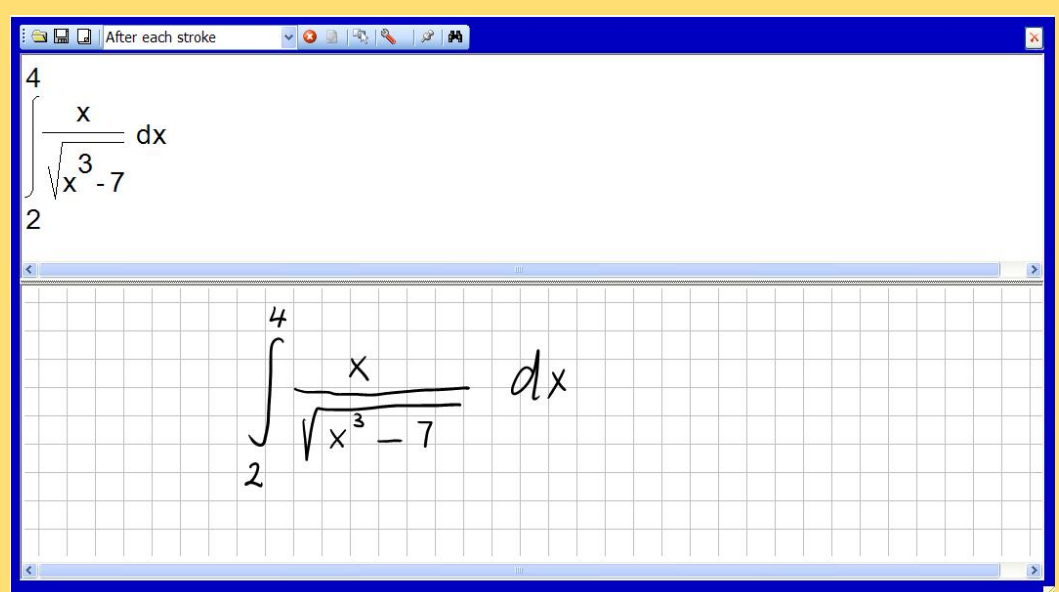
Toward math-literate computers

Scott MacLean, George Labahn, Edward Lank, Mirette Marzouk, David Tausky





The case for pen-based mathematics

We are working on MathBrush - a pen-based system for doing mathematics.

Compare  with `\int_{-2}^4 \frac{x}{\sqrt{x^3-7}} dx`

Which is easier? More natural?

Consider evaluating the integral: 

Would you work this out by hand? 

Math literacy and its challenges

A literate system (Blostein 2009):

- possesses a significant and flexible understanding of its domain
- can communicate effectively with the user, at the user's level of understanding

Literacy is a useful guiding principle for designing MathBrush.

It is difficult to achieve. How can a computer:

- interpret complex two-dimensional sketches?
- deal with ambiguous handwriting and notation?
- understand the semantics and intent of writing to communicate effectively?

We explore some of these questions and offer some preliminary suggestions.

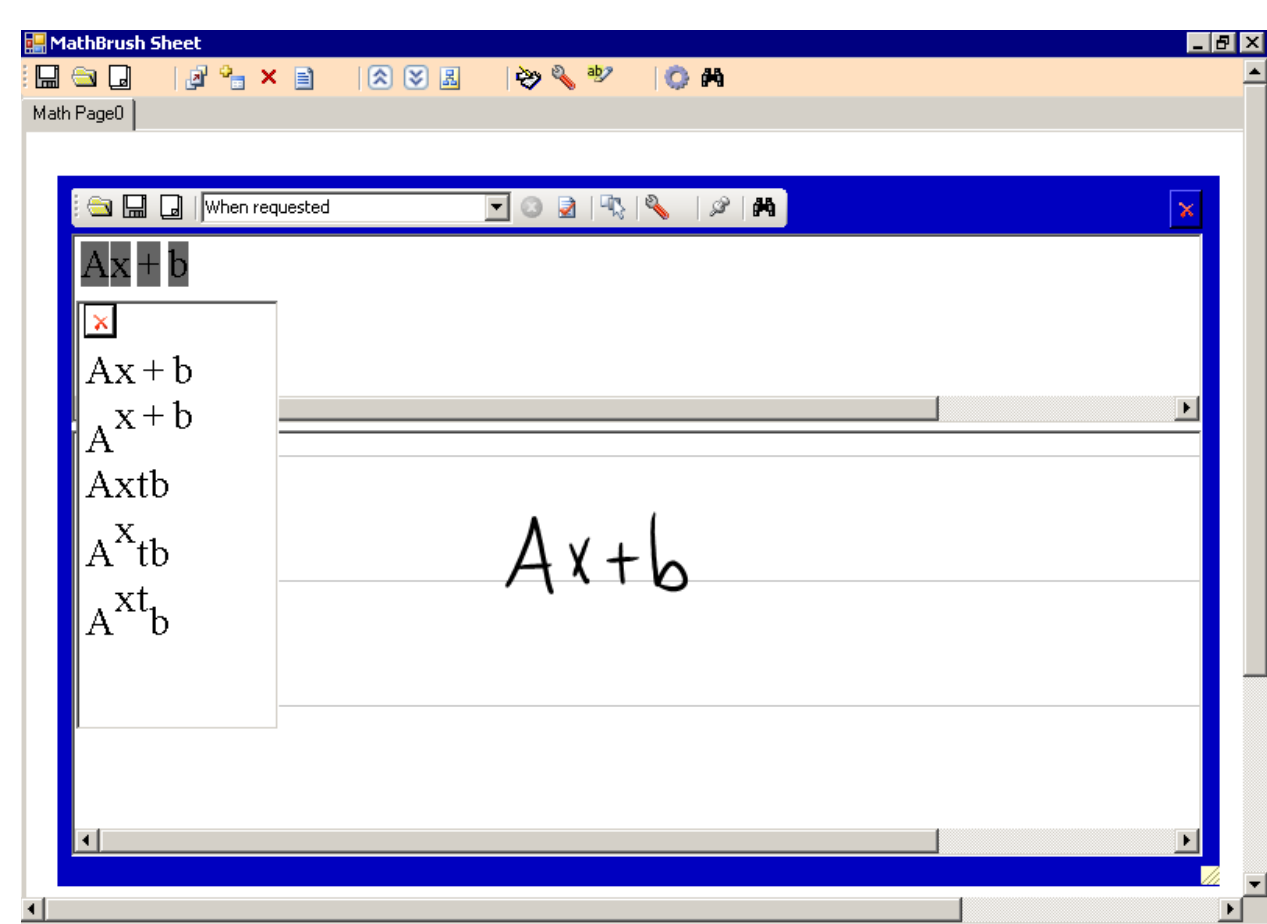
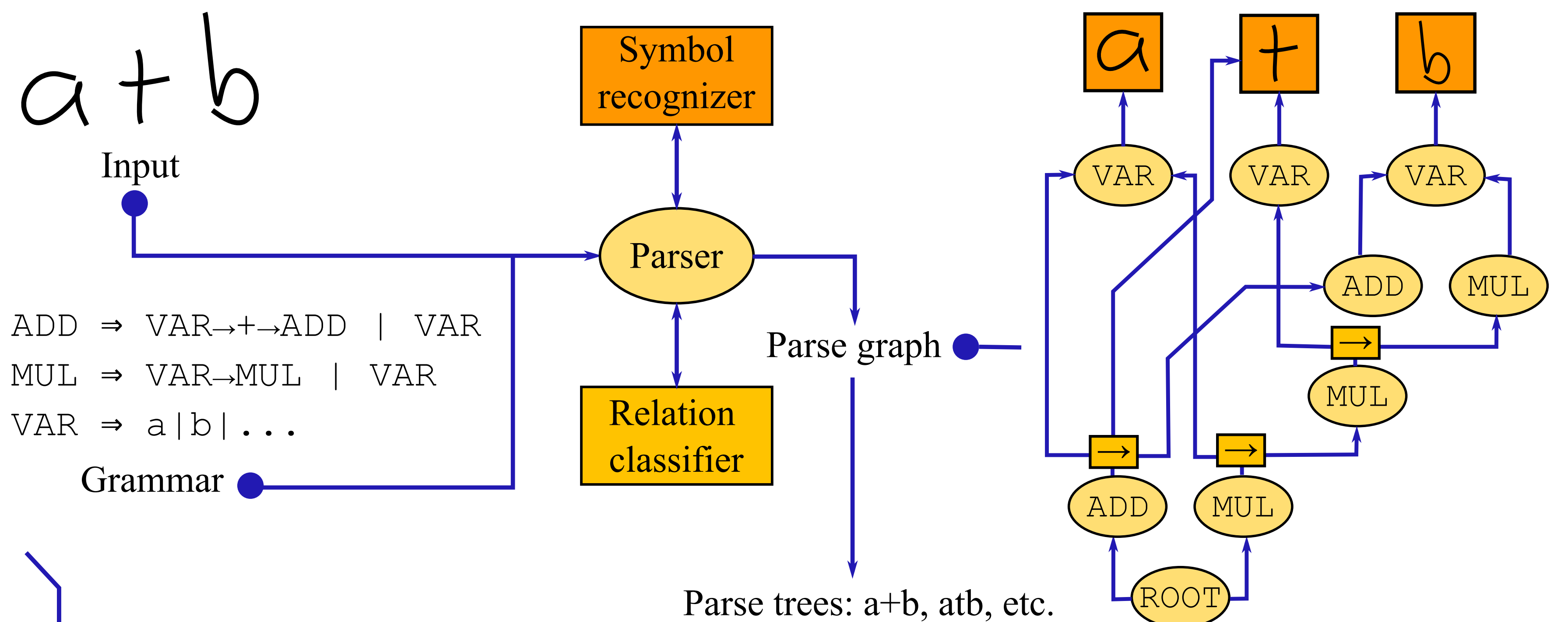
Parsing two-dimensional input

Combine a grammar with a formal model of recognition ambiguity

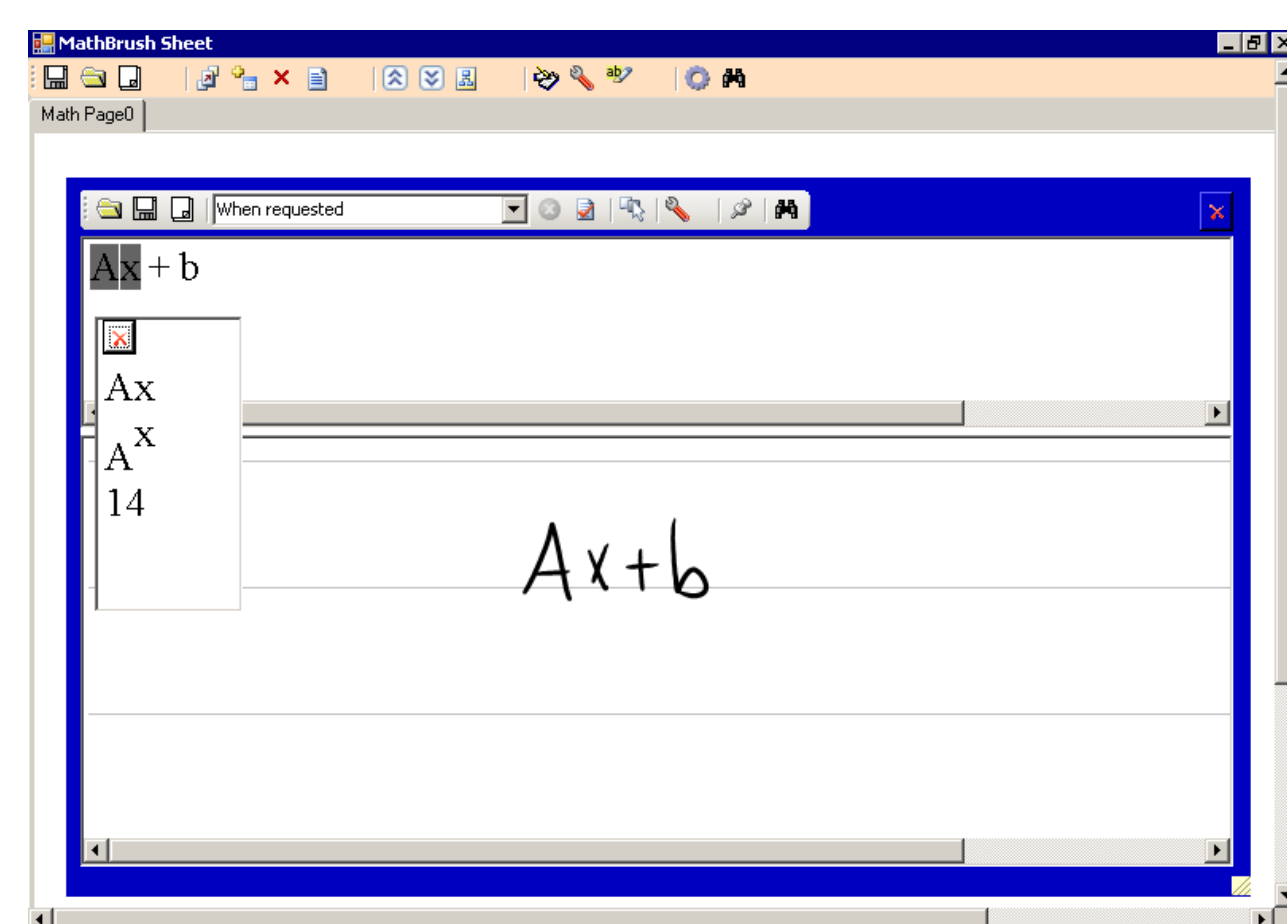
- 2-D notation modeled by spatial relations between subexpressions
- ambiguity modeled by fuzzy set theory

To parse:

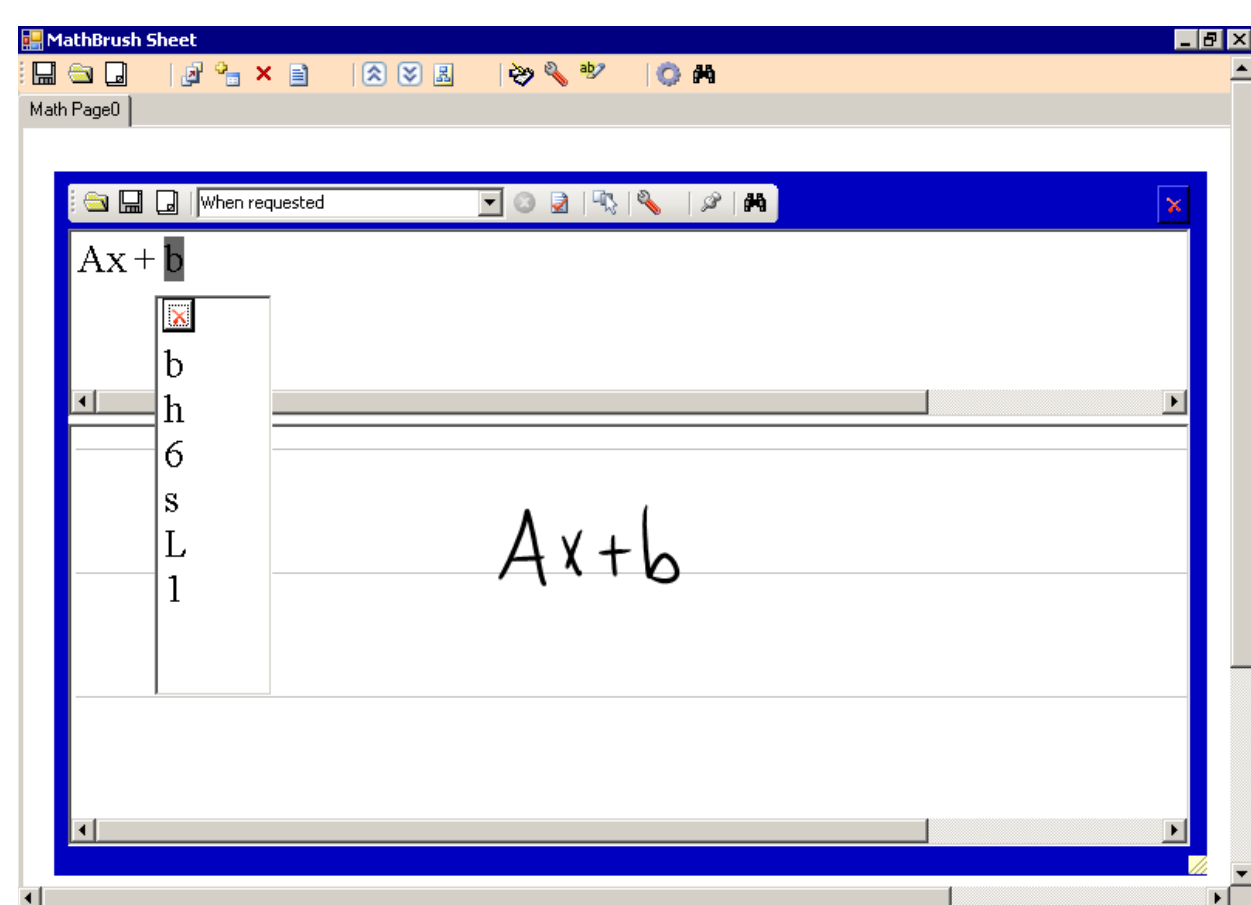
1. Build a graph modeling all possible parses
2. Extract parses in decreasing order of parse confidence



Alternates for the entire expression



Alternates for a subexpression



Alternates for a terminal symbol

Evaluating system literacy

- MathBrush offers a simple correction facility in case of recognition errors.

Instead of counting correctly-recognized expressions...

Test how hard it is for the user to get the job done! E.g.,

- how many corrections were required?
- how many incorrect results were seen?
- how long did it take?

We automated this process: Given an input expression,

1. Run the recognizer
2. Simulate the corrections a user would make
3. Count them

Data for training and testing

For automatic training and testing, we need many handwritten expressions.

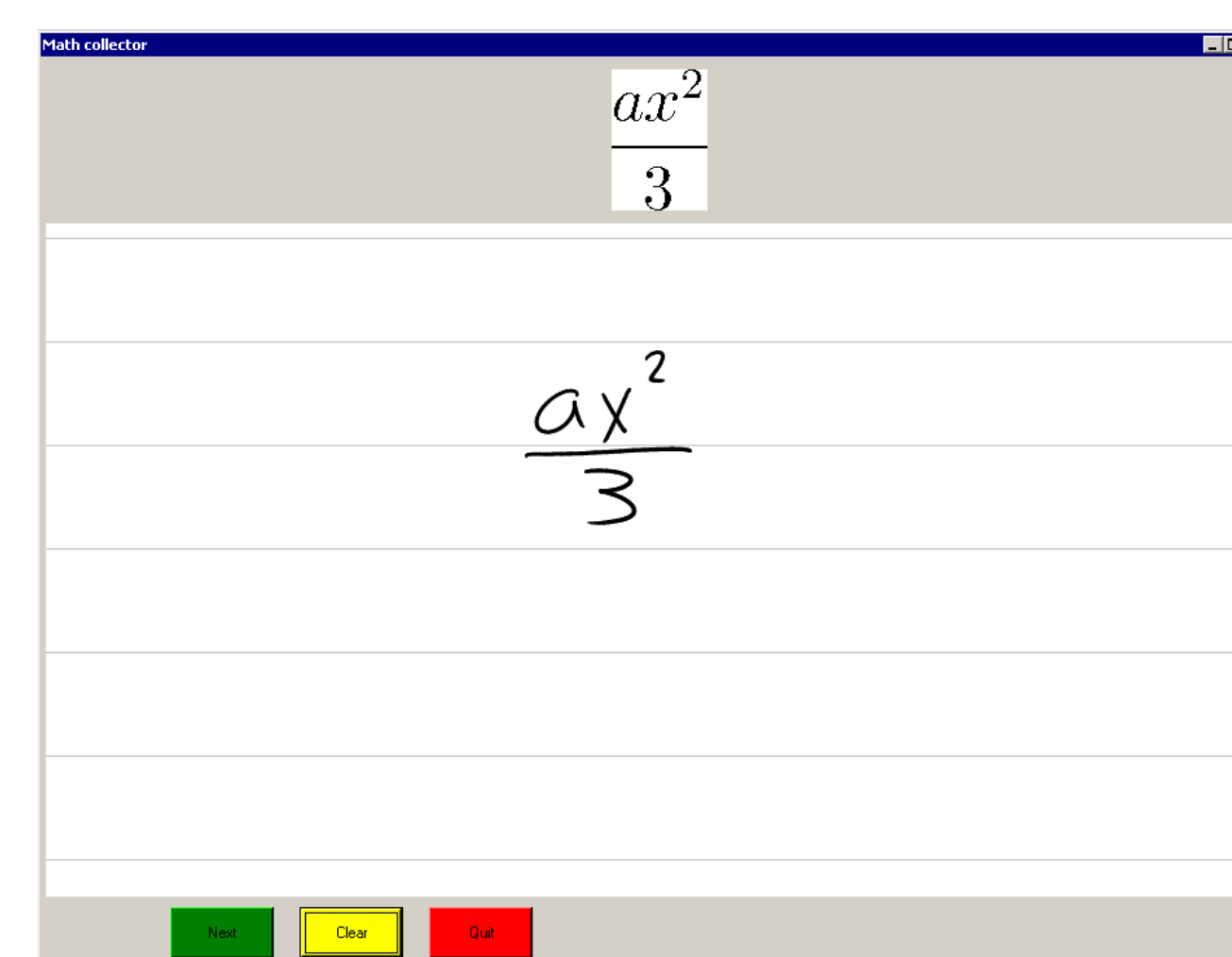
Collected with the help of automatic tools and undergrad students.

Process:

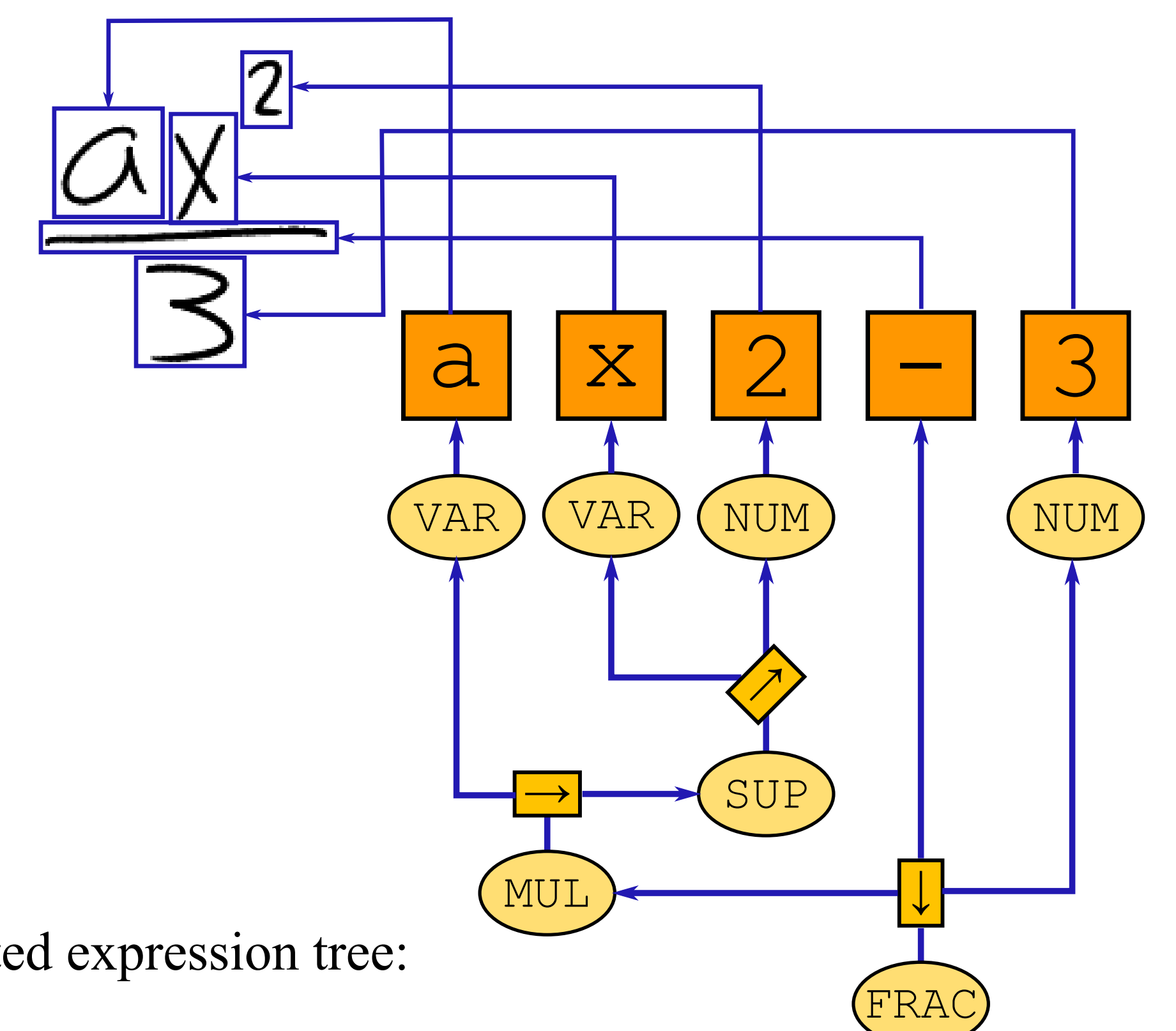
1. Derive a random expression in the grammar model.
2. Convert it to an image and display it to a participant for transcription.
3. Automatically label the drawing with ground-truth
 - match symbols between derived and transcribed expression
 - accuracy over 99%

We collected over 5000 expressions.

Available from <http://www.cs.uwaterloo.ca/scg/mathbrush/corpus>



Custom software for ink collection



Represented as an annotated expression tree:

What's next?

More mathematical understanding - matrices, ellipses (...), typed variables, ...

More extensible syntax

More user studies

See <http://www.cs.uwaterloo.ca/scg/mathbrush>