Multi-Domain Sketch Recognition

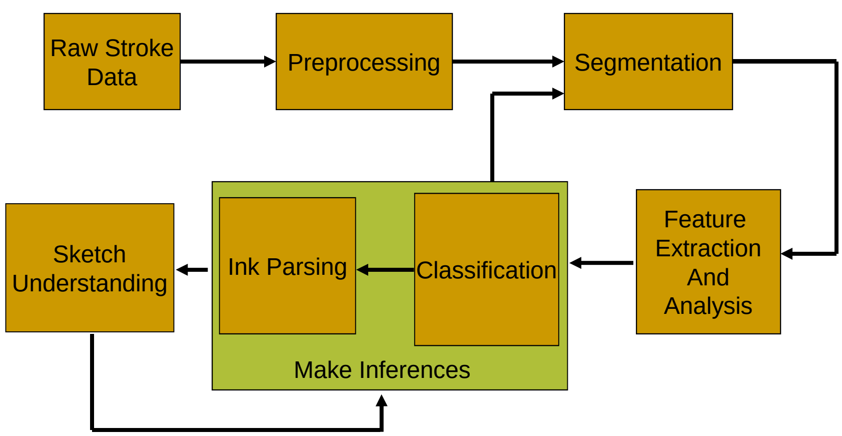
Lecture #11: Sketch Understanding

Joseph J. LaViola Jr.

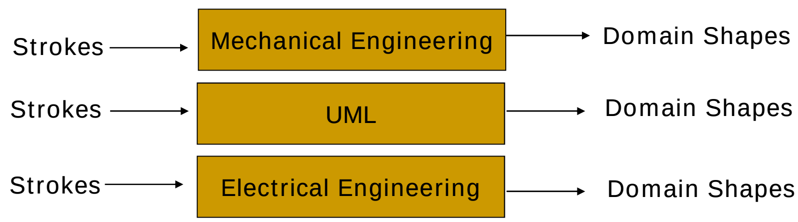
Fall 2010

Slides adapted from Alvarado, Multi-Domain Sketch Understanding, SIGGRAPH course #3, 2007.

Recall Pen-Based Interface Dataflow

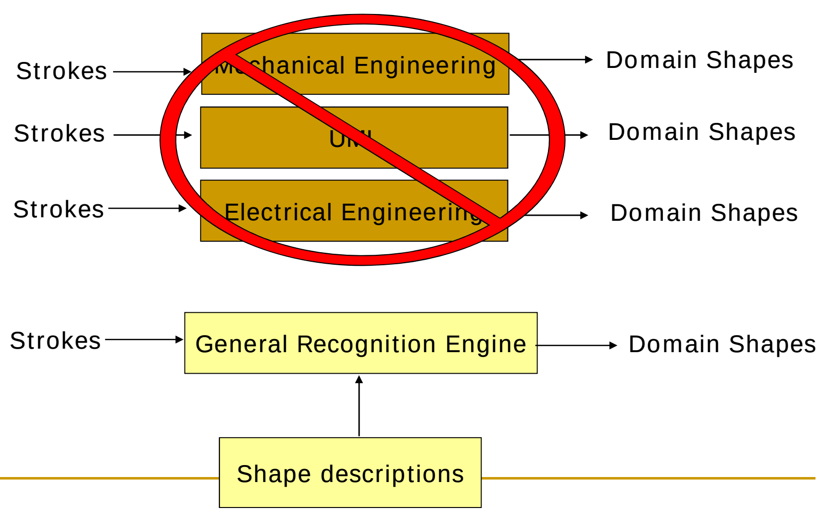


Building Recognition Systems



- Building each system requires:
  - sketch recognition expertise
  - a lot of time (2-5 person years!)
  - built in domain assumptions to improve recognition

A Multi-Domain Sketch Recognition Engine

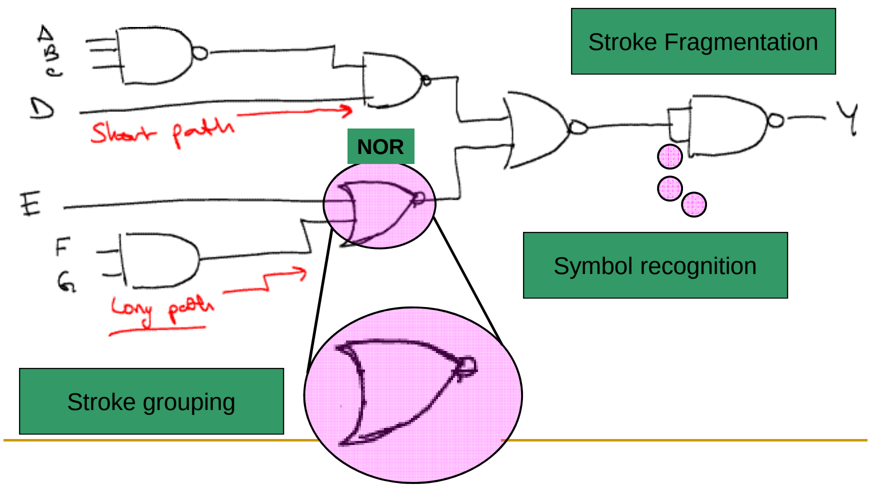


Enabling Natural Interaction

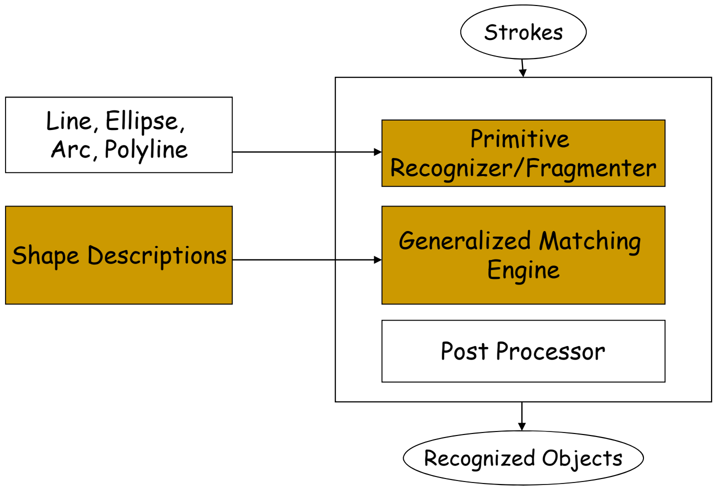
- Goal:
  - recognition engines for multiple domains
- Core challenge:
  - multi-domain recognition

Sketch Recognition Subtasks

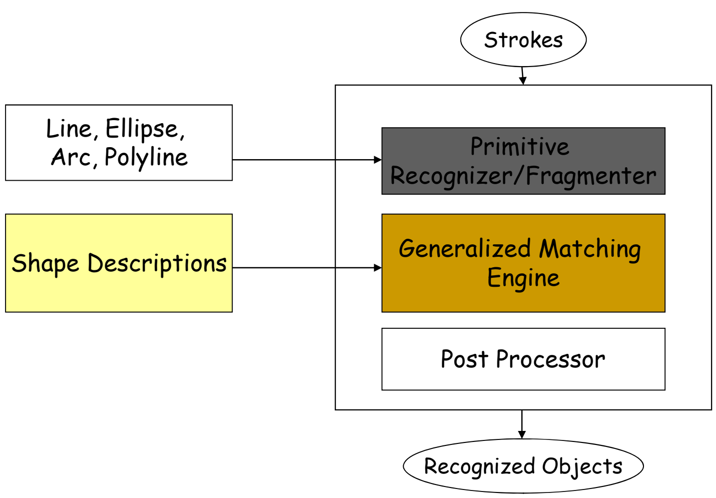
- Need a multi-domain solution!



Multi-Domain Sketch Recognition Architecture

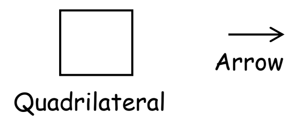


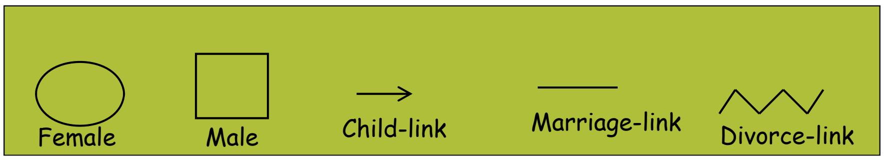
Multi-Domain Sketch Recognition Architecture



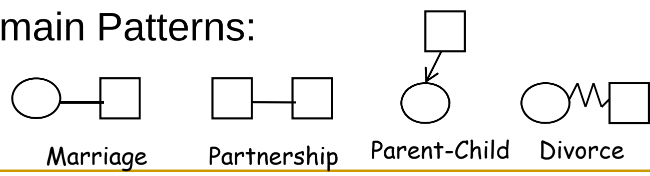
Family Tree Domain

- Compound:
- Domain:

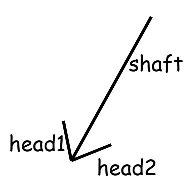




- Domain Patterns:



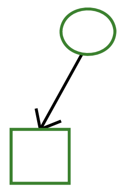
Knowledge Representation (LADDER [Hammond03])



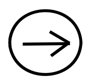
Shape defined by
Subshapes
Constraints

```
(Define Arrow
  (Subshapes (Line shaft)
             (Line head1)
             (Line head2))
  (Constraints
    (coincident shaft.p1 head1.p1)
    (coincident shaft.p1 head2.p1)
    (equalLength head1 head2)
    (smaller head1 shaft)
    (acuteAngle head1 shaft)
    (acuteAngle head2 shaft)))
```

Knowledge Representation

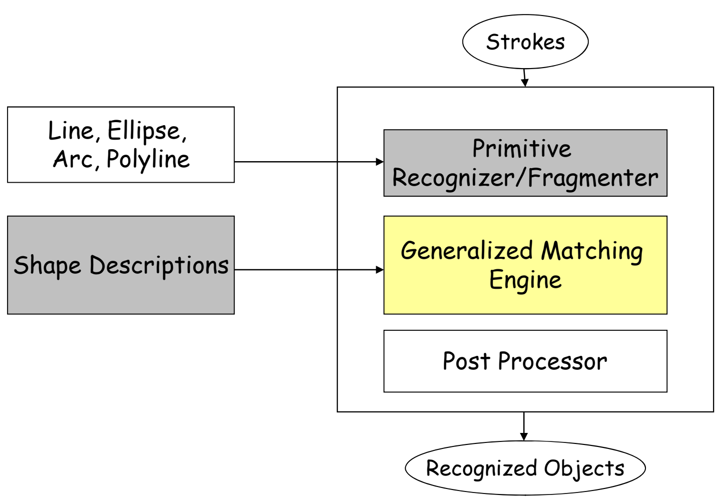


```
(Define Child-link
 (Subshapes (Arrow a)))
```



```
(Define Current-Source
 (Subshapes (Arrow a)
            (Ellipse e))
 (Constraints
   (contains e a)))
```


Multi-Domain Sketch Recognition Architecture

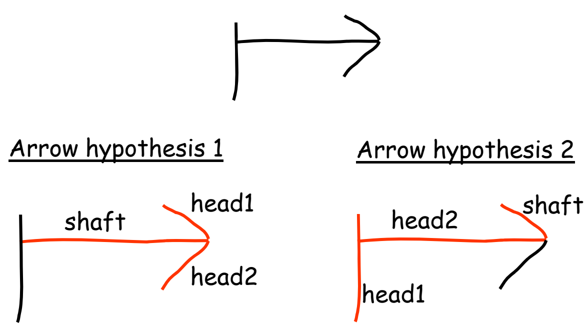


Recognition overview

- Task: Simultaneous fragmentation, grouping and symbol identification
- Constraint-based approach
- Generate and test

Definition

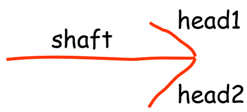
- *<u>Hypothesis</u>:* A shape description with associated mapping from subshapes to user's strokes.



Fall 2010 CAP 6105 – Pen-Based User Interfaces ©Joseph J. LaViola Jr.

Hypothesis-based recognition

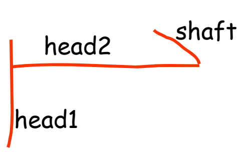
- Given a hypothesis, determine if it matches a shape description by testing constraints




```
(Define Arrow
 (Subshapes (Line shaft)
            (Line head1)
            (Line head2))
 (Constraints
  (coincident shaft.p1 head1.p1)
  (coincident shaft.p1 head2.p1)
  (equalLength head1 head2)
  (smaller head1 shaft)
  (acuteAngle head1 shaft)
  (acuteAngle head2 shaft)))
```


Hypothesis-based recognition

- Given a hypothesis, determine if it matches a shape description by testing constraints




```
(Define Arrow
 (Subshapes (Line shaft)
            (Line head1)
            (Line head2))
 (Constraints
  (coincident shaft.p1 head1.p1)
  (coincident shaft.p1 head2.p1)
  (equalLength head1 head2)
  (smaller head1 shaft)
  (acuteAngle head1 shaft)
  (acuteAngle head2 shaft)))
```

Hypothesis-based recognition: Issues

- Too many hypotheses to try them all

$$\sum_{i \in S} \binom{n}{k_i} (k_i!)$$

n = number of strokes;
S = set of shapes;
k_i = subcomponents in shape S_i

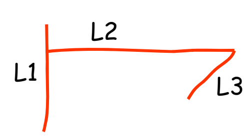
- Constraints depend on context

And this only considers shapes *independently!*

Definition

- *<u>Partial Hypothesis</u>:* A hypothesis with unbound subshapes

<u>Quadrilateral partial hypothesis</u>

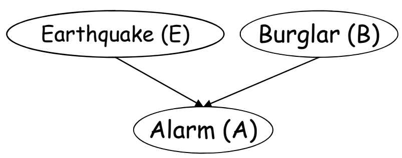


L4 is unbound

Recognition Using Partial Hypotheses

- Generating Hypotheses (rule-based)
  - generate partial hypotheses (PHs) based on easily recognizable low-level shapes
  - fill in strong PHs with unrecognized strokes
  - prune weak PHs
- Evaluating Hypotheses (probabilistic)
  - how well do user's strokes fit low level shapes?
  - how well are constraints satisfied?

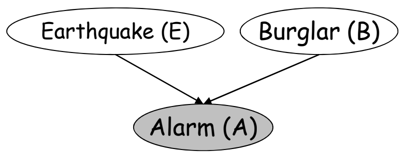
Bayesian Networks [Pearl88]



Use Bayes Rule to reason about the certainty of each variable

- Reason about events/entities
- Two parts
  - directed Acyclic Graph:
    - assign meaning to nodes
    - specify which variables influence one another
  - conditional Probability Tables
    - specify *how* variables influence one another

Bayesian Networks [Pearl88]



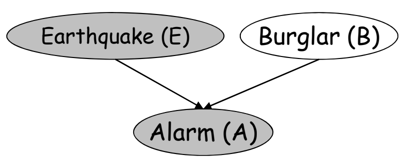
- Observations give evidence for other variables

  Say we observe A=t, then

  P(E|A)=0.0056

  P(B|A)=0.49

Bayesian Networks [Pearl88]



- Observations give evidence for other variables

  Say we observe A=t, then

  P(E|A)=0.0056

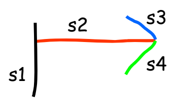
  P(B|A)=0.49
- Important Phenomenon: Explaining away

  If we *also* hear there has been an earthquake (i.e., E=t), then

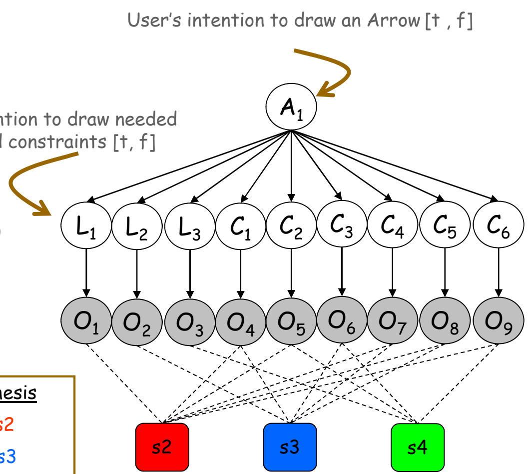
  P(B|A,E) = 0.001

Shape Fragments

(Define Arrow
(**Subshapes**
L_1: (Line shaft)
L_2: (Line head1)
L_3: (Line head2))
(**Constraints**
C_1: (coincident shaft.p1 head1.p1)
C_2: (coincident shaft.p1 head2.p1)
C_3: (equalLength head1 head2)
C_4: (smaller head1 shaft)
C_5: (acuteAngle head1 shaft)
C_6: (acuteAngle head2 shaft)))

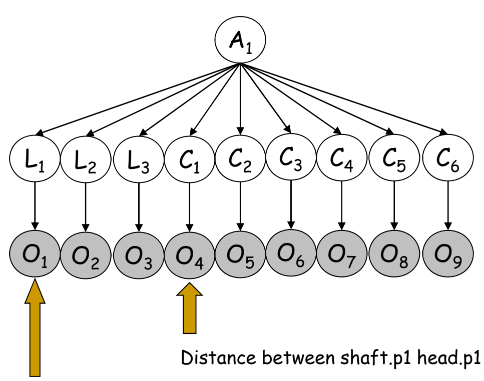


Arrow Hypothesis
shaft = s2
head1 = s3
head2 = s4



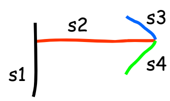
Shape Fragments: Measurement Nodes

(Define Arrow
(**Subshapes**
L_1: (Line shaft)
L_2: (Line head1)
L_3: (Line head2))
(**Constraints**
C_1: (coincident shaft.p1 head1.p1)
C_2: (coincident shaft.p1 head2.p1)
C_3: (equalLength head1 head2)
C_4: (smaller head1 shaft)
C_5: (acuteAngle head1 shaft)
C_6: (acuteAngle head2 shaft)))

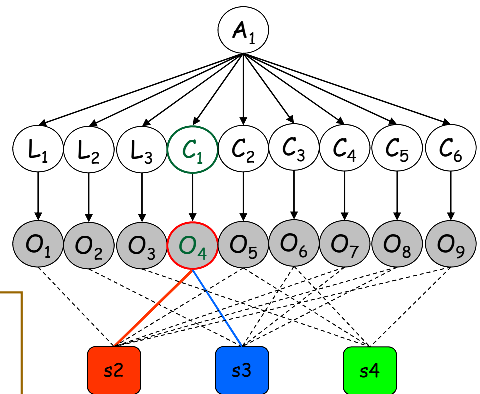


Shape Fragments

(Define **Arrow**
(**Subshapes**
L_1: (Line shaft)
L_2: (Line head1)
L_3: (Line head2))
(**Constraints**
C_1: (coincident shaft.p1 head1.p1)
C_2: (coincident shaft.p1 head2.p1)
C_3: (equalLength head1 head2)
C_4: (smaller head1 shaft)
C_5: (acuteAngle head1 shaft)
C_6: (acuteAngle head2 shaft)))

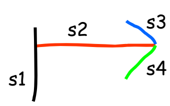


Arrow Hypothesis
shaft = s2
head1 = s3
head2 = s4

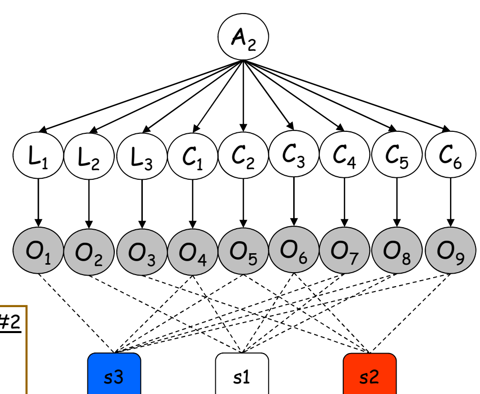


Shape Fragments: Another Hypothesis

(Define **Arrow**
(**Subshapes**
L_1: (Line shaft)
L_2: (Line head1)
L_3: (Line head2))
(**Constraints**
C_1: (coincident shaft.p1 head1.p1)
C_2: (coincident shaft.p1 head2.p1)
C_3: (equalLength head1 head2)
C_4: (smaller head1 shaft)
C_5: (acuteAngle head1 shaft)
C_6: (acuteAngle head2 shaft)))

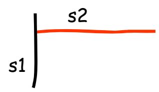


Arrow Hypothesis #2
shaft = s3
head1 = s1
head2 = s2

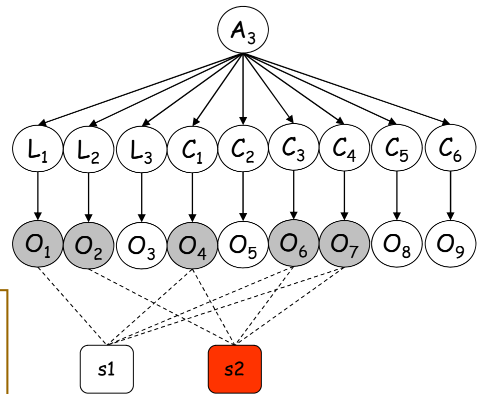


Shape Fragments: Partial Hypothesis

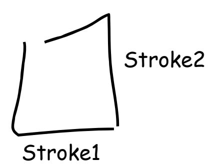
(Define **Arrow**
(**Subshapes**
L_1: (Line shaft)
L_2: (Line head1)
L_3: (Line head2))
(**Constraints**
C_1: (coincident shaft.p1 head1.p1)
C_2: (coincident shaft.p1 head2.p1)
C_3: (equalLength head1 head2)
C_4: (smaller head1 shaft)
C_5: (acuteAngle head1 shaft)
C_6: (acuteAngle head2 shaft)))

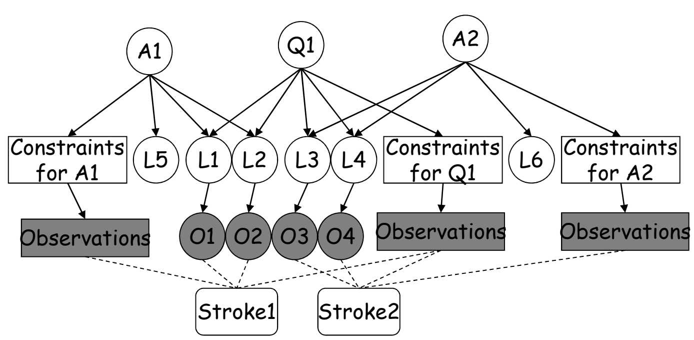


Arrow Hypothesis #3
shaft = s1
head1 = s2
head2 = ??



Composing Shape Fragments



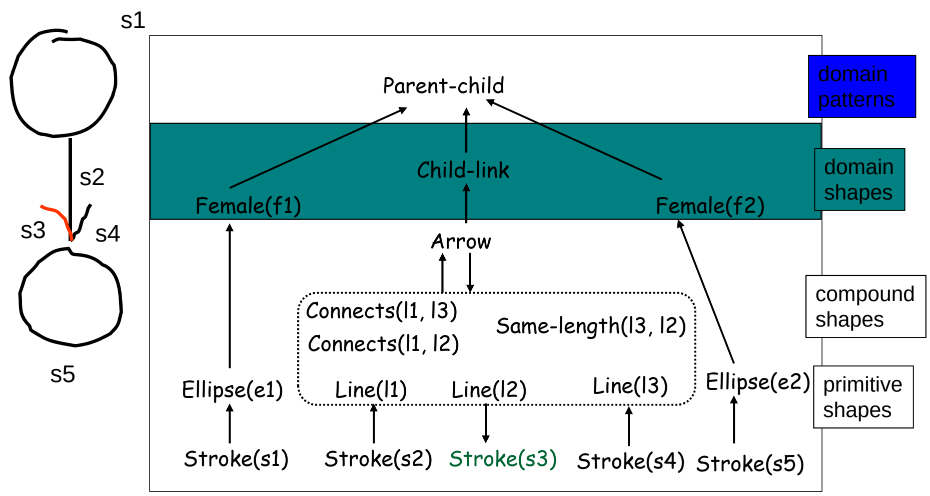


Each node represents a hypothesis

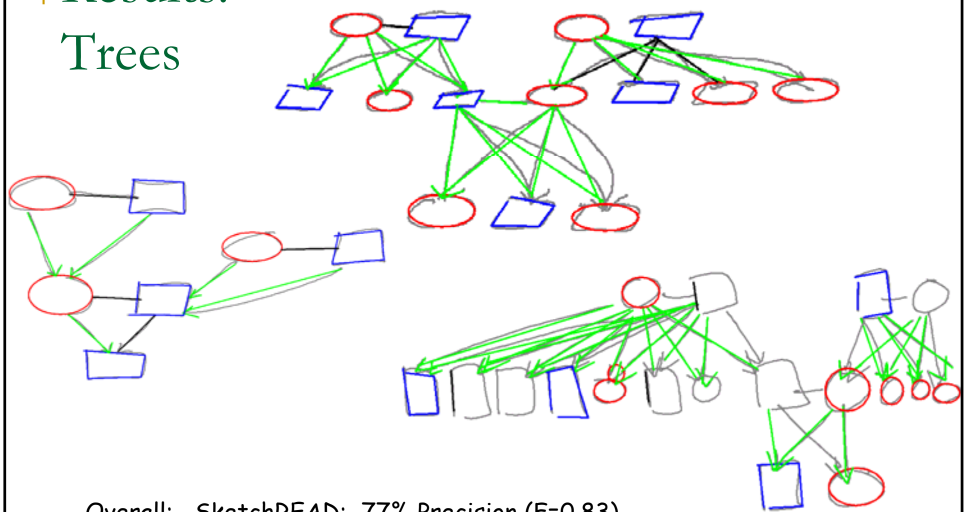
Hypothesis Generation

- Bottom Up
  - partial hypotheses generated based on rough classification for objects and constraints
- Top Down
  - strokes possibly reclassified to fit into PHs
- Pruning
  - keep number of hypotheses manageable

An Illustration

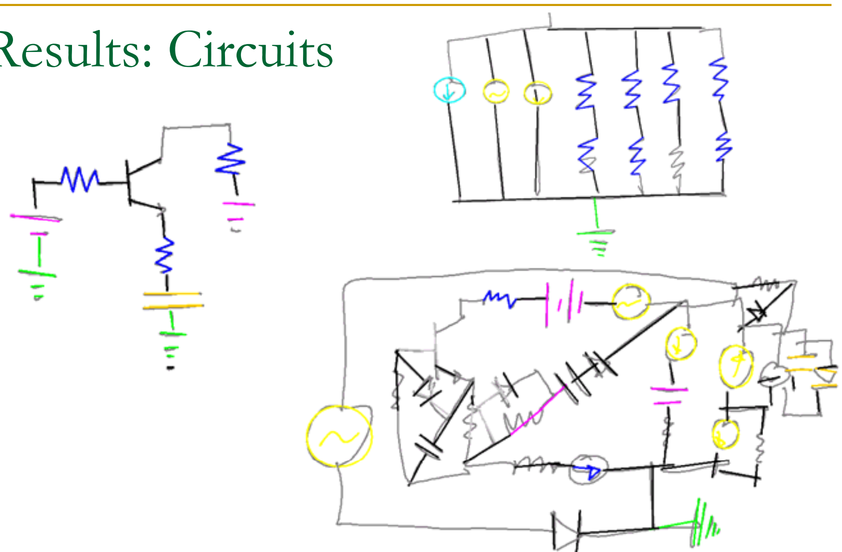


Results: Trees



Overall: SketchREAD: 77% Precision (F=0.83)
Baseline: 50% Precision (F=0.65)

Results: Circuits



Overall: SketchREAD: 62% Precision (F=0.65)
Baseline: 54% Precision (F=0.57)

Readings

- LaViola, J. and Zeleznik, R. MathPad^2: A System for the Creation and Exploration of Mathematical Sketches" *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 23(3):432-440, August 2004.
- Christine Alvarado and Randall Davis. SketchREAD: A Multi-Domain Sketch Recognition Engine. In *Proceedings of UIST 2004*, pp.23-32. New York, New York, October 24-27 2004.
- Lockwood, K., Lovett, A., Forbus, K., Dehghani, M., and Usher, J. Automatic Interpretation of Depiction Conventions in Sketched Diagrams. *Proceedings of the Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 167-174, 2008.
- Hammond, T., and R. Davis. Ladder: A Sketching Language for User Interface Developers, *Computers and Graphics 29*, 518-532, 2005.