Lecture 1
CS 1813 – Discrete Mathematics

Learning Goals
Lesson Plans
and
Logic

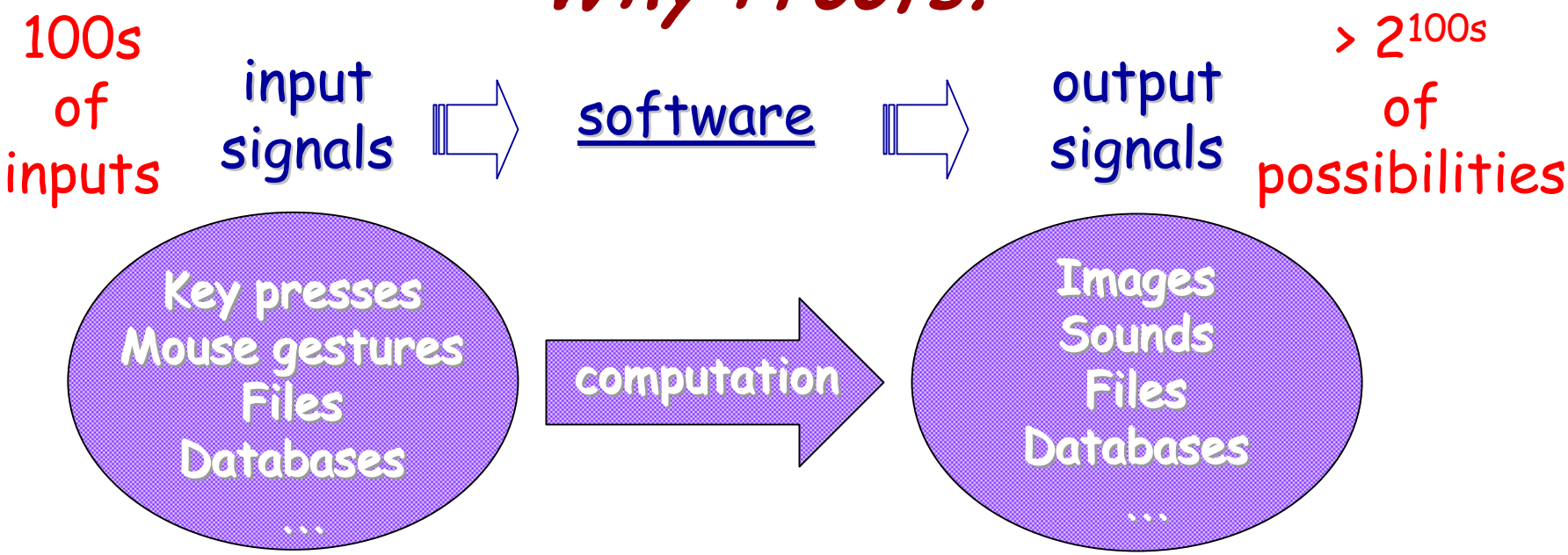
Rex Page
Professor of Computer Science
University of Oklahoma
EL 119 – Page@OU.edu

Learning Goals

- **Apply mathematical logic to prove properties of software**
  - ✓ Predicate calculus and natural deduction
  - ✓ Boolean algebra and equational reasoning
  - ✓ Mathematical induction
  - ✓ Mathematical induction
  - ✓ Mathematical induction
- **Understand fundamental data structures**
  - ✓ Sets
  - ✓ Trees
  - ✓ Functions and relations
- Additional topics
  - ✓ Graphs
  - ✓ Counting
  - ✓ Algorithm Complexity

proofs galore!
proofs galore!
proofs galore!
proofs galore!
proofs galore!
proofs galore!

Why Proofs?



- ✓Software translates input signals to output signals
- ✓A program is a constructive proof of a translation
- ✓But what translation?
- ✓Proofs can confirm that software works correctly
- ✓Testing cannot confirm software correctness
- ✓Practice with proofs improves software thinking

Textbook and Tools

- *Discrete Mathematics Using a Computer*

  Cordelia Hall and John O'Donnell

  Springer-Verlag, January 2000

- Tools provided with textbook
  - Download from course website for CS 1813
- Hugs interpreter for Haskell
  - Download from course website
  - Haskell is a math notation (and a programming lang)
- Reading assignments begin with Chapter 2
  - Read Chapter 1 (Haskell) as needed, for reference
  - Haskell coverage JIT, like other math notations

Formal Mathematical Notations

- Notations introduced as needed (JIT)
  - Logic $a \wedge b$, $a \vee b$, $a \rightarrow b$, $\forall x.P(x)$, $\exists x.Q(x)$, ...
  - Sets $A \cup B$, $A \cap B$, $\{x \mid x \in S, P(x)\}$, ...
  - Sequences

Haskell

```
[x | x <– s, P(x)]
[4, 7, 2] ++ [3, 7] == [4,7,2,3,7]
s(a: xs) = s[x | x <– xs, x < a]
           ++ [a] ++
           s[x | x <– xs, x >= a]
```

  - Structures

```
Theorem [P, Q] (And P Q)
```

Coursework

Class Attendance REQUIRED

- Reading assignments
  - ✓See syllabus on course website
  - ✓Study prior to class
- Class Participation — 10%
- Homework problem sets
  - ✓Approximately weekly — 10%
- Midterm Exam 1 — 20%
- Midterm Exam 2 — 20%
- Final Exam — 40%

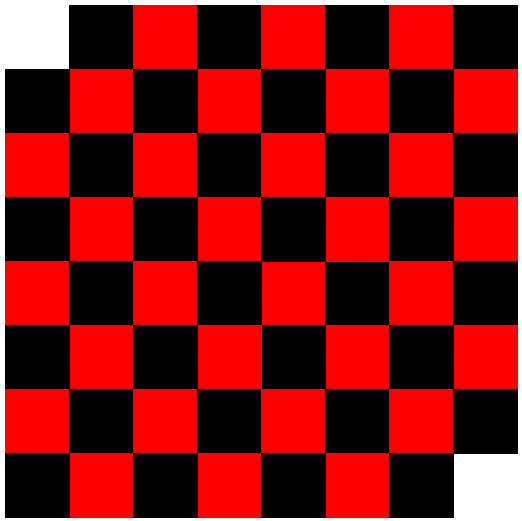
Contribution to grade

Q/A Lab – Thursdays 8:00pm, CEC 439

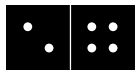
Q/A Lab Attendance NOT REQUIRED

Tiling with Dominos

a mathematical proof – just for practice



checkerboard with
two missing corners



Dominos – size matches board

Problem

- cover board with dominos
- no overlapping dominos
- no dominos outside board

✓How many squares on board?

✓So, how many dominos will it take?

✓One domino covers how many red squares?

✓31 dominos cover how many red squares?

✓How many red squares are there?

✓Yikes! What's wrong here?

TILT

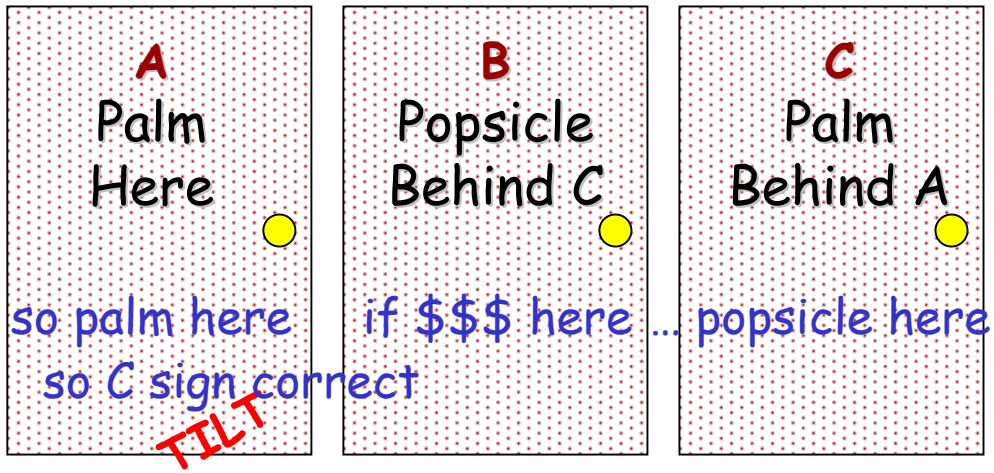
Adapted from Singh, *Fermat's Enigma*, Walker & Co, 1997

How To Find a Million Dollars

using logic

Three Doors

- ✓ Behind one is a million dollars
- ✓ Behind another is a Palm Pilot
- ✓ Behind the other is a melting Popsicle



Where's the jackpot?

- Why not A?
- Why not B?
- Must be C, eh?

Bonus question:

Where's the Palm Pilot?

- Door C speaks the truth – the Palm Pilot is behind A
- Door B lies – it has a Popsicle, afterall

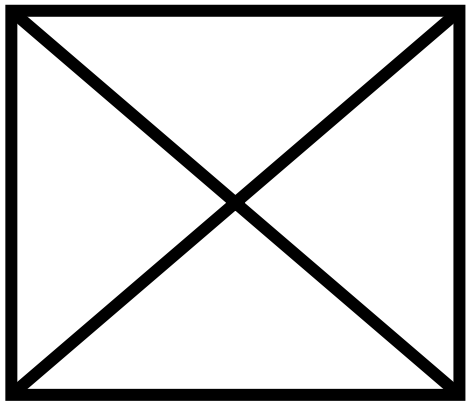
Signs on Doors

- ✓$$$ door: true statement
- ✓ Popsicle door: false statement

If it was so, it might be;
and if it were so, it would be:
but as it isn't, it ain't. That's logic.
- Tweedledee
in Through the Looking Glass

Adapted from Smullyan, *The Lady or the Tiger*, Times Books, 1982

Tracing a Square and Its Diagonals



Square + Diagonals

Problem

- Start at any corner
- Trace some line to another corner
- Then trace from **that** corner to another
- Keep going until all six lines are traced
- Don't trace any line more than once
  (crossing OK, but not retracing)

Solution revealed in the next lecture

Homework #1

- Problem under "Assignments" tab in course website
- It's a hard problem
- You don't have much mathematical apparatus, yet, to attack it
- Grade based more on thoughtfulness and well-expressed ideas than on solutions

End of Lecture 1