Univeral Turing Machines and Diagonalization

Mark Greenstreet, CpSc 421, Term 1, 2006/07

- Universal Turing Machines
  - A Turing Machine that can be programmed to simulate any other Turing Machine.
- Diagonalization
  - A way to show compare the sizes of infinite sets.
  - On Monday, we'll use it to give a formal proof that the Halting Problem is undecidable.

Some “Universal” Languages

- $A_R = \{D\#w \mid D$ describes a DFA that accepts string $w\}$
  - This is the “universal” language for Regular Languages.
  - We described a Turing Machine for A_R in the Nov. 1 lecture.
- $A_{CFL} = \{G\#w \mid G$ describes a CFG that generates string $w\}$
  - This is the “universal” language for Context-Free Languages.
  - We described a Turing Machine for A_{CFL} in the Nov. 1 lecture.
- $A_{TM} = \{M\#w \mid M$ describes a TM that accepts string $w\}$
  - This is the “universal” language for Turing Recognizable Languages.
  - We’ll described a Turing Machine for A_{TM} now.

A Universal Turing Machine

$A_{TM} = \{M\#w \mid M \text{ describes a TM that accepts string } w\}$

We'll define a Turing Machine, U, that recognizes A_{TM}.

Σ_U: $\{\mathtt{0}, \mathtt{1}, \mathtt{(}, \mathtt{,}, \mathtt{)}, \mathtt{\#}\}$

Γ_U: $\Sigma \cup \{\Box, \ldots\}$

w: The format for the input tape is described on the next slide.

Tapes: We'll use six tapes:

- $input$: The input string, $M\#w$ is written here.
- δ_M: A list of tuples representing the transition function of M is written here.
- q_M: The current state of M is written here.
- c_M: The current tape symbol of M is written here.
- $tape_M$: The current tape contents for M.
- $scratch$: A scratch tape.

Input Tape Format for U

$|Q_M|$`,`$|\Sigma_M|$`,`$|\Gamma_M|\delta_M$`#`w where

$|Q_M|$: Binary representation of the number of states of M.

$|\Sigma_M|$: Binary representation of the number of symbols in the input alphabet of M.

$|\Gamma_M|$: Binary representation of the number of symbols in the tape alphabet of M.

δ_M: A list of tuples for the transition function for M. Each tuple has the form: (q, c, q', c', d) where $\delta_M(q, c) = (q', c', d)$. In other words, when M is in state q and reads c, it transitions to state q', writes a c' on the tape and moves one square in direction d, $d \in \{\mathtt{0}, \mathtt{1}\}$, where `0` denotes a left move and `1` denotes a right move.

q_0, $accept$, and $reject$: we assume that these special states are represented by 0, 1, and 2 respectively.

w: The input string: binary numbers separated by commas. We assume that each symbol in Γ is encoded using the same number of bits, $\lceil \log_2 |\Gamma| \rceil$.

Operation of U (1/2)

Make sure the input is valid:

- Check that the tape has the form $B^*\texttt{,}B^*\texttt{,}B^*C^*\texttt{\#}B^*(\texttt{,}B^*)^*$ where

  $B = \{\texttt{0}, \texttt{1}\}$

  $C = \texttt{(}B^*\texttt{,}B^*\texttt{,}B^*\texttt{,}B^*\texttt{,}B^*\texttt{)}$

  Note: This format requirement is a regular language. U can check this by scanning the tape from left-to-right using its finite states to implement a DFA.

- Read $|Q_D|$, $|\Sigma_D|$ and Γ_D.
- Copy δ_M onto the δ_M tape.
- Make sure that each tuple, (q, c, q', c', d) for δ_M has $q, q' \in 0 \ldots (|Q_D| - 1)$, $c, c' \in 0 \ldots (|\Gamma_D| - 1)$, $d, \in B$. Make sure all combinations for q and c are covered.
- Copy w onto the $tape_M$ tape.
- Make sure that each symbol for w is in Σ_D.

Operation of U (2/2)

- Simulate M.

```
q → 0
while(q ∉ {1,2}) {
  c → string in B* under head on tape_M.
  scan δ_M tape to find entry for (q,c),
    let this be (q,c,q',c',d)
  copy q' onto the q tape.
  copy c' onto the tape_M tape.
  move head for tape_M according to d.
}
if(q == 1) accept;
else reject.
```

Observations

- If M accepts w, then U accepts $M\#w$.
- If M rejects w, then U rejects $M\#w$.
- If M loops on w, then U loops on $M\#w$.
- $\therefore U$ recognizes A_{TM}.
- U is universal:
  - One machine U works with any input $M\#w$.
    In other words, U can simulate any Turing machine, M.
  - You can think of the M part of $M\#w$ as a program, and the w part as the input data for the program.
  - U is a programmable machine. Rather than building a new TM for each problem, we just program U to simulate whatever TM we want.

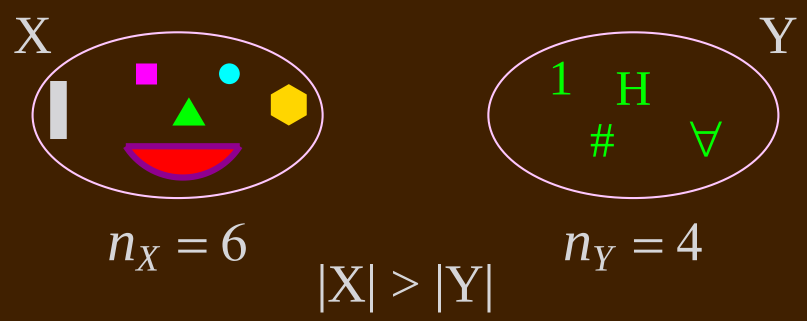
The Halting Problem for Turing Machi

- From the previous slide, U loops on input $M\#w$ iff M loops on input w.
- We've shown that U recognizes A_{TM}, but it doesn't decide A_{TM}.
- Could we build some other machine, U' that can determine when a machine M loops on its given input? If so, then U' would decide A_{TM}.
  - This would require solving the Halting Problem for Turing Machines.begin
  - We gave an informal argument (see the Oct. 23 slides) that the Halting Problem for Java$^{\text{TM}}$ programs is undecidable (by Java programs). On Monday (Nov. 6), we'll show that the Halting Problem for Turing Machines is undecidable.
  - First, we'll look at "diagonalization", the main mathematical concept that we'll need for the proof.

Which Set is Bigger?

- Let X and Y be sets.
- Is $|X| > |Y|$?
- Solution by counting:
  - Count each element in X. Let n_X be the number.
  - Count each element in Y. Let n_Y be the number.
  - If $n_X > n_Y$, then $|X| > |Y|$.

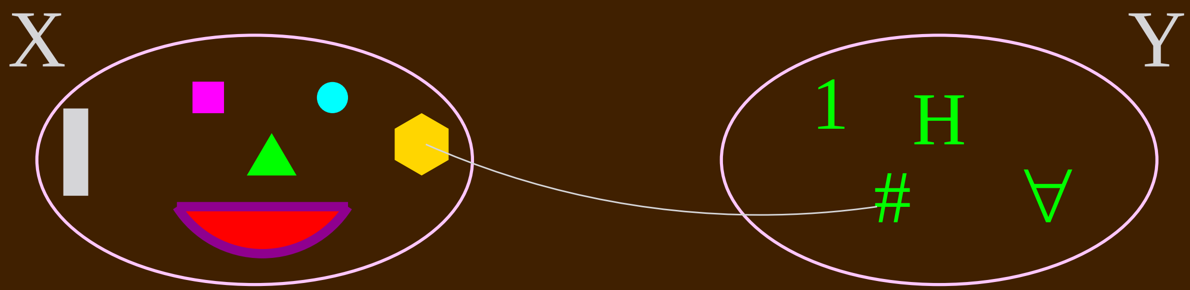
Which Set is Bigger?



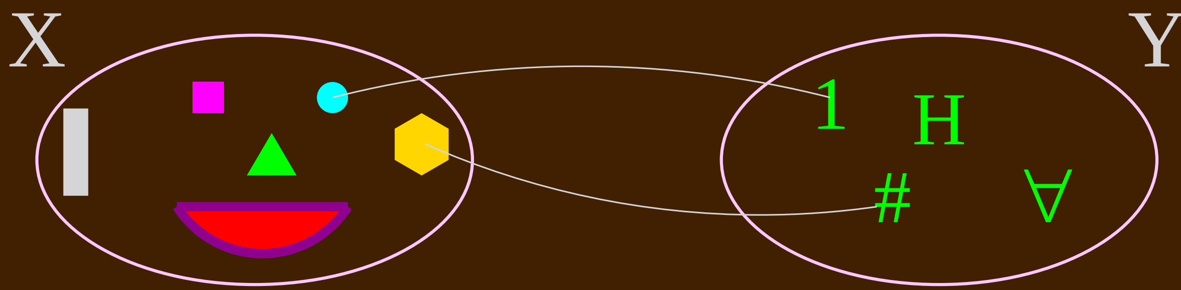
- Let X and Y be sets.
- Is $|X| > |Y|$?
- Solution by counting:
  - Count each element in X. Let n_X be the number.
  - Count each element in Y. Let n_Y be the number.
  - If $n_X > n_Y$, then $|X| > |Y|$.

Comparing by Pairing

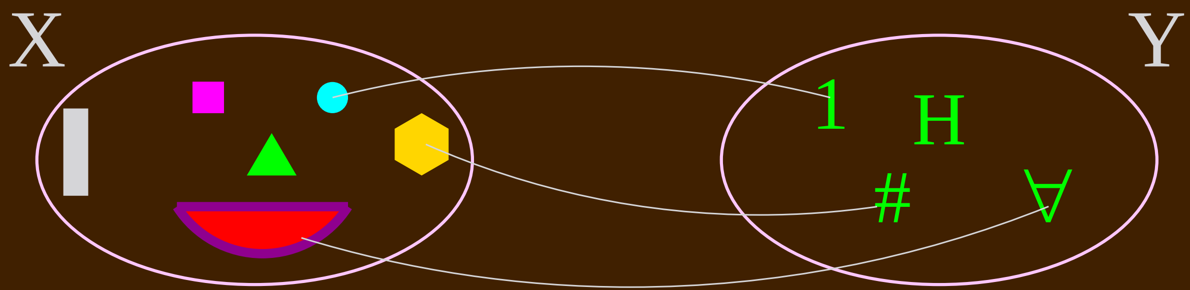
Comparing by Pairing



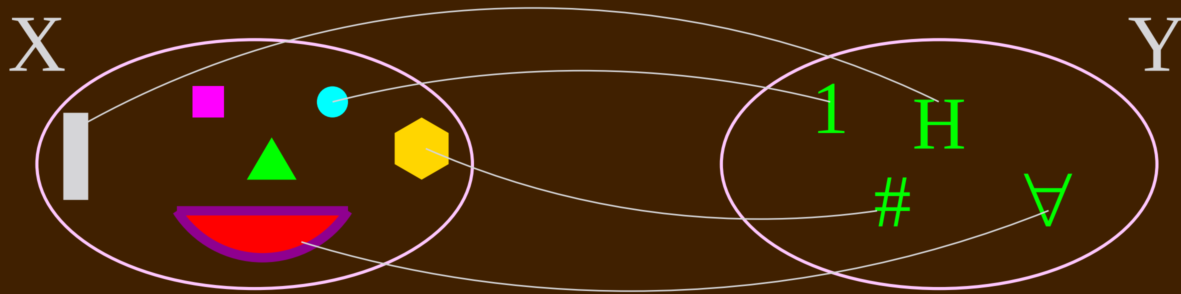
Comparing by Pairing



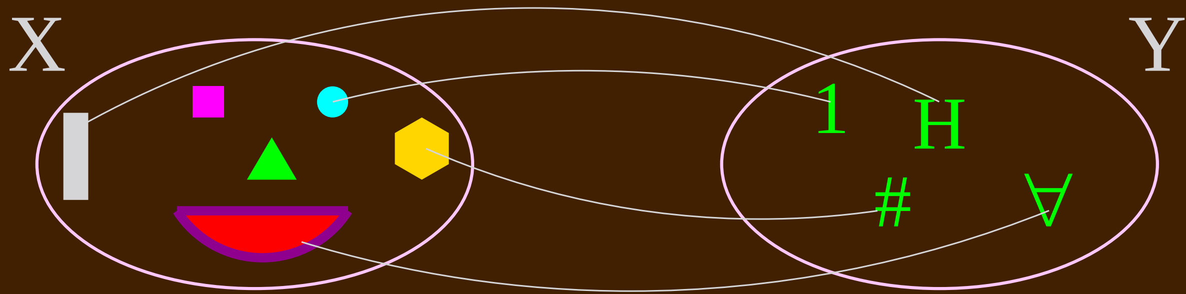
Comparing by Pairing



Comparing by Pairing



Comparing by Pairing



$$|X| \geq |Y|$$

- If there is an onto function, $f : X \rightarrow Y$, then $|X| \geq |Y|$.
- If there are onto functions $f : X \rightarrow Y$ and $g : Y \rightarrow X$, then $|X| \geq |Y|$ and $|Y| \geq |X|$, Thus, $|X| = |Y|$.
- Note that if $f : X \rightarrow Y$ is one-to-one and onto, then f^{-1} exists and is one-to-one, and onto as well. Thus, if there is a one-to-one and onto function, $f : X \rightarrow Y$, then $|X| = |Y|$.

Even Integers vs. All Integers

- Let $\mathbb{Z}$ be the set of all integers, and $\mathbb{E}$ be the set of all even integers.
  - Let $f : \mathbb{Z} \rightarrow \mathbb{E}$ be the function $f(x) = 2x$.
  - f is one-to-one: If $f(x) = f(y)$, then $2x = 2y$, and $x = y$.
  - f is onto: If $y \in \mathbb{E}$, then $y/2 \in \mathbb{Z}$, and $f(y/2) = y$.
  - $\therefore$ $\mathbb{E} = \mathbb{Z}$.

    In English, this says that the number of even integers is equal to the number of all integers!
- A similar argument shows that $|\mathbb{N}| = |\mathbb{Z}|$.

Naturals vs. Rationals

- Let $\mathbb{Q}^+$ be the set of all strictly-positive rational numbers, and $\mathbb{N}^+$ be the strictly-positive naturals.
- Let $f : \mathbb{Q}^+ \rightarrow \mathbb{N}^+$ with $f(x) = \lceil x \rceil$. Clearly, f is onto, thus $|\mathbb{Q}^+| \geq |\mathbb{N}^+|$ — there are at least as many positive rational numbers as positive naturals.
- Let $g : \mathbb{Q}^+ \rightarrow \mathbb{N}^+$ with

$$\begin{aligned}
g(n) &= \tfrac{x(n)+1-zn}{z(n)} \\
x(n) &= \left\lfloor \tfrac{1}{2}(\sqrt{8n-7}+1) \right\rfloor \\
y(n) &= \tfrac{1}{2}(x(n)^2 - x(n)) \\
z(n) &= n - y(n)
\end{aligned}$$

For example:

n	1	2	3	4	5	6	7	8	9	10	11	...
$x(n)$	1	2	2	3	3	3	4	4	4	4	5	...

Reading List:

- Today: Sipser, 4.2
- Nov. 6: Sipser, 4.2 (cont., midterm 2 cutoff)
- Nov. 8: Sipser, 5.1
- Nov. 10: Sipser, 5.1 (cont.)
- Nov. 13: Remembrance Day (no lecture)
- Nov. 15: Midterm 2
- Nov. 17: Sipser, 5.2
- Nov. 20: Sipser, 5.3