Lecture 3

Pete Manolios
Northeastern

Aunt

Who knows what an aunt is?

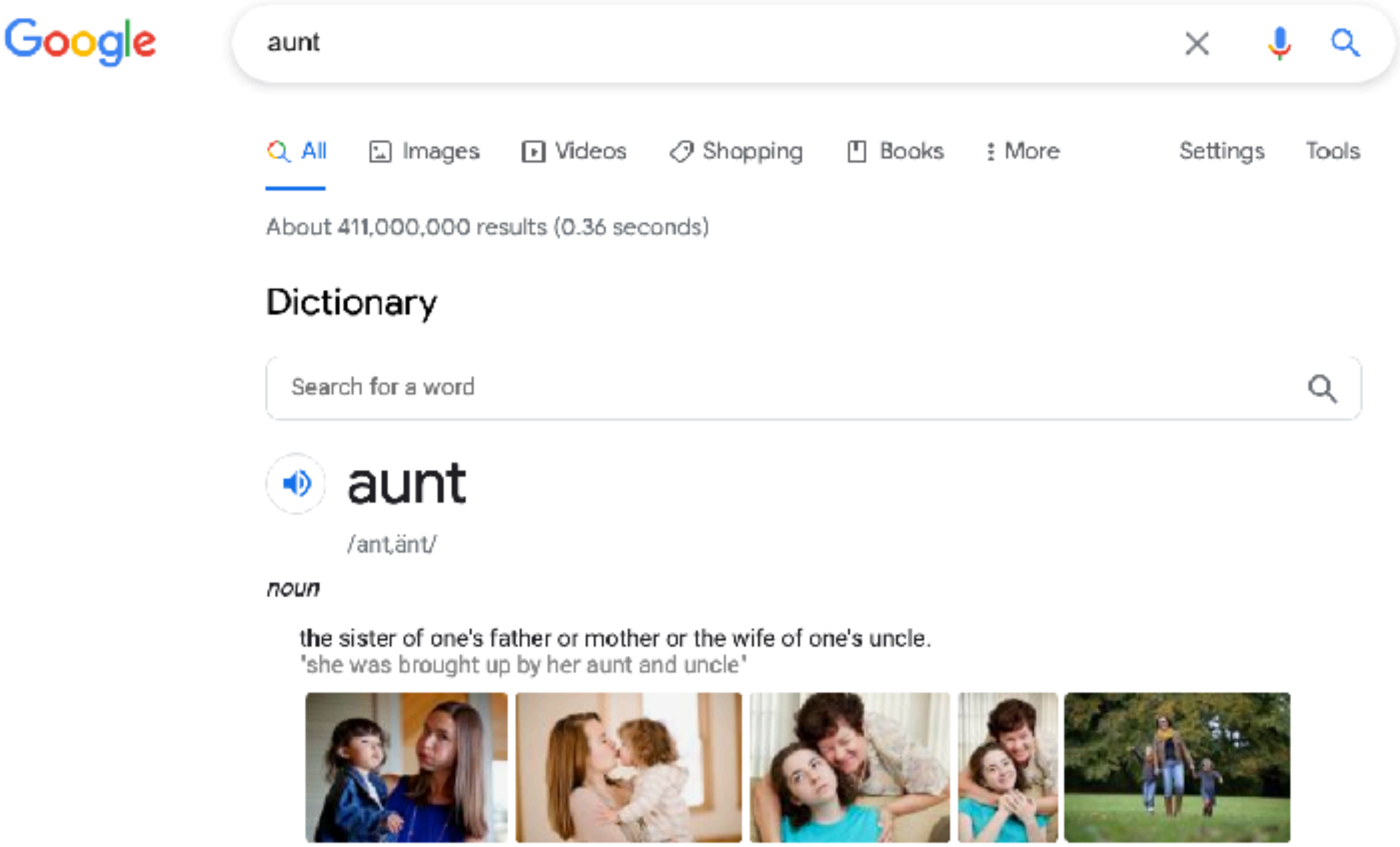
An aunt is …

Aunt May Parker

Real Name	May Parker
First Appearance	Amazing Fantasy #15 (August 1962)

Slides by Pete Manolios for CS4820

Hey, Google



aunt

All Images Videos Shopping Books More Settings Tools

About 411,000,000 results (0.36 seconds)

Dictionary

Search for a word

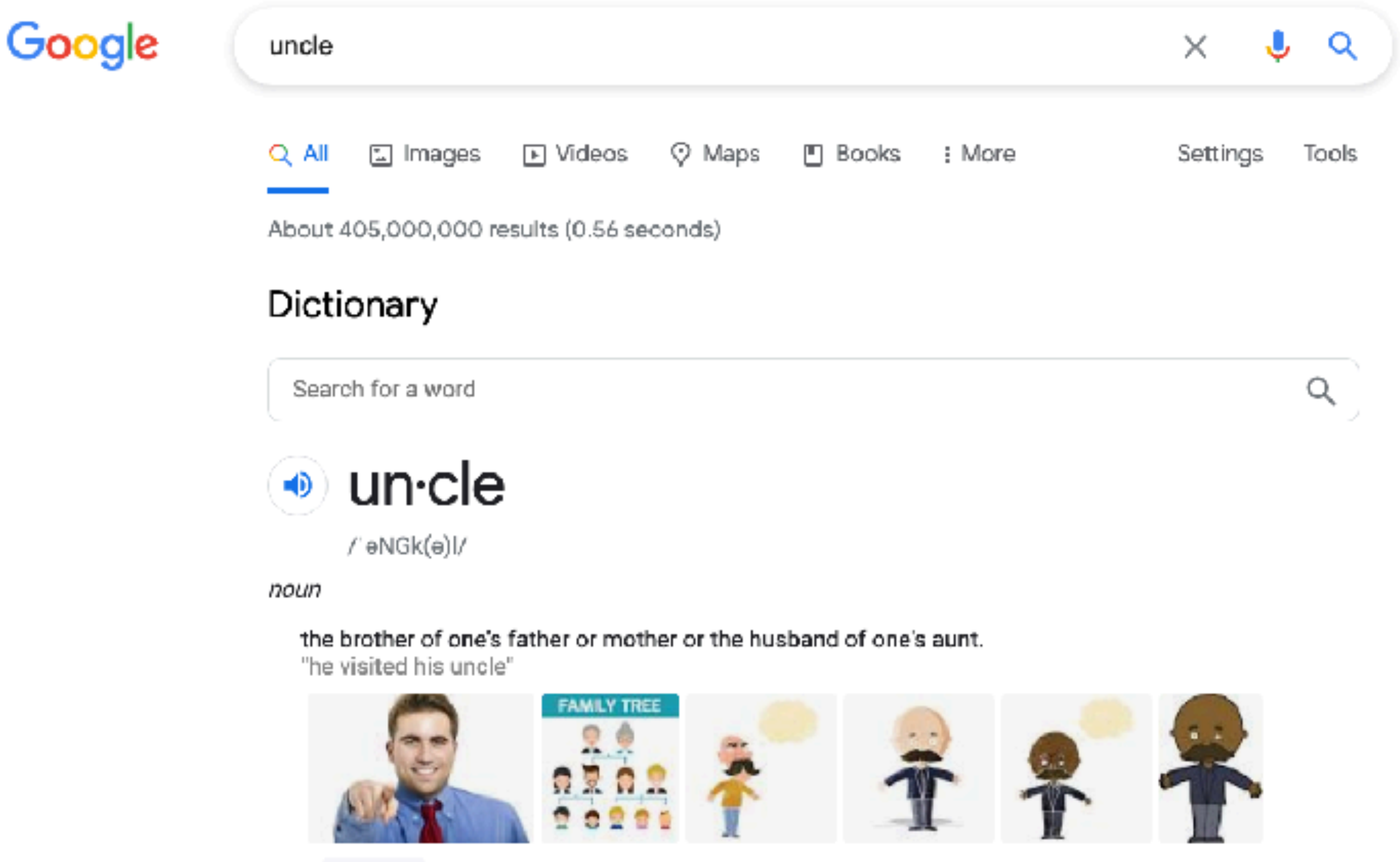
aunt

/ant,änt/

noun

the sister of one's father or mother or the wife of one's uncle.
'she was brought up by her aunt and uncle'

Hey, Google



Google

uncle

All · Images · Videos · Maps · Books · More · Settings · Tools

About 405,000,000 results (0.56 seconds)

Dictionary

Search for a word

un·cle

/ˈəNGk(ə)l/

noun

the brother of one's father or mother or the husband of one's aunt.
"he visited his uncle"

Aunts and Definitions

- The world according to Goggle
  - Aunt: The sister of one’s father or mother or the wife of one’s uncle.
  - Uncle: The brother of one’s father or mother or the husband of one’s aunt.
  - Google’s definition is circular! So, it is unknowable and requires trust in Google.
- Ignore the circularity for a moment; any other issues?
- What if your mother’s sister is married to a woman?
  - Not an aunt according to Google!
- Wikipedia: An aunt is a woman who is a sibling of a parent or married to a sibling of a parent.
  - What is a sibling? (Adopted? Half?)
  - What about parent? (Biological?)
  - Married? (Legal marriage? What about divorce?)
  - Temporal aspect? (Sure, can’t guess the future)
- Property: If *X* is the aunt of *Y*, then *Y* is not the aunt of *X*. True or Not??
- Logic, mathematics, reasoning requires *real* definitions, allowing *real* inferences

Formalizing Aunts

$$\frac{\alpha P \beta \quad \alpha P x \quad F x \quad \beta P y}{x A y}$$

$$\begin{tikzcd}
\alpha \arrow[d] \arrow[dr] & \\
\beta \arrow[d] & x \\
y &
\end{tikzcd}$$

P: parent-of F: female A: aunt-of

⭕ Female ○ Unspecified sex

Slides by Pete Manolios for CS4820

Falsifying the Property

Property: If *X* is the aunt of *Y*, then *Y* is not the aunt of *X*.

$$\frac{\alpha P\beta \quad \alpha Px \quad Fx \quad \beta Py}{xAy} \qquad \frac{\gamma P\delta \quad \gamma Py \quad Fy \quad \delta Px}{yAx}$$

Instantiate

$$\begin{tikzcd}
\alpha \arrow[d] \arrow[dr] & \\
\beta \arrow[d] & x \\
y &
\end{tikzcd}
\qquad
\begin{tikzcd}
\gamma \arrow[d] \arrow[dr] & \\
\delta \arrow[d] & y \\
x &
\end{tikzcd}$$

Merge

$$\begin{tikzcd}
 & \gamma \arrow[d] \arrow[dddl, bend right] \\
\alpha \arrow[d] \arrow[dr] & \delta \arrow[d] \\
\beta \arrow[d] & x \\
y &
\end{tikzcd}$$

P: parent-of F: female A: aunt-of

⭕ Female ◯ Unspecified sex

Pro Reasoning Tip:
Make your own notations &
patterns: xAy is x, 1up, 2down

Slides by Pete Manolios for CS4820

Definitions

Mathematics is entirely free in its development, and its concepts are only linked by the necessity of being consistent, and are co-ordinated with concepts introduced previously by means of precise definitions.

Georg Cantor

If any philosopher had been asked for a definition of infinity, he might have produced some unintelligible rigmarole, but he would certainly not have been able to give a definition that had any meaning at all.

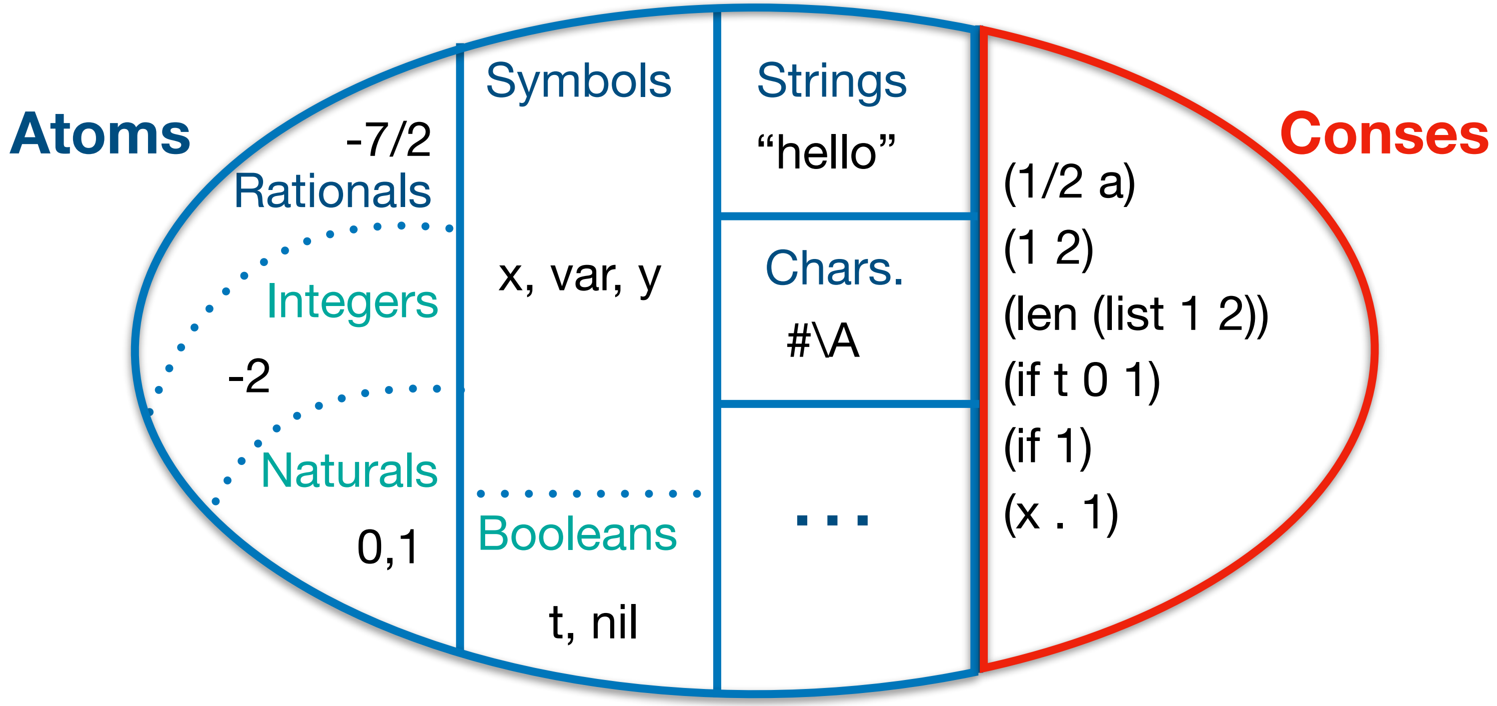
Bertrand Russell

The deepest definition of youth is life as yet untouched by tragedy.

Alfred North Whitehead

ACL2 Universe (Review)

All = Conses ∪ **Atoms**



Lists = Conses ∪ { ( ) }

True-lists = $\cup_{i \in \mathbb{N}} L_i$

L_0 = { ( ) }, L_{i+1} = L_i ∪ {`(cons x l): x ∈ All, l ∈ L`$_i$}

Slides by Pete Manolios for CS4820

Expressions

- "Expressions" (or "terms") are elements of a subset of U (the Universe)
- Evaluation maps expressions to ACL2 objects
- ⟦*expr*⟧ denotes the semantics of *expr*
  - or what *expr* evaluates to at the REPL
- Constants are expressions that evaluate to themselves
  - `⟦t⟧ = t`
  - `⟦nil⟧ = nil`
  - `⟦6⟧ = 6`
  - `⟦-21⟧ = -21`

Lazy vs Strict

- Semantics of `if`
  - ⟦(`if` *test then else*)⟧ = ⟦*then*⟧ , when ⟦*test*⟧ ≠ `nil` (Generalized Booleans)
  - ⟦(`if` *test then else*)⟧ = ⟦*else*⟧ , when ⟦*test*⟧ = `nil`
- `if` is lazy:
  - first ACL2s evaluates *test*, i.e., it computes ⟦*test*⟧
  - if ⟦*test*⟧ ≠ `nil` then ACL2s returns ⟦*then*⟧
  - otherwise, it returns ⟦*else*⟧
- So, *test* is always evaluated, but only one of *then*, *else* is
- All other functions are strict
  - ACL2s evaluates all of the arguments to the function
  - Then ACL2s applies the function to evaluated results

Function Definitions

- Why does this definition make sense?
- Because it terminates
- A key idea every time you define a program is to convince yourself that on every recursive call, some parameter decreases in a well-founded way
- Hmm, can lists be circular? then what?
- Lists are non-circular in ACL2s, which is why this works
- Termination is one of the key ideas in CS
- Note that data driven definitions always terminate

```
(definec mlen (l :tl) :nat
  (if (endp l)
      0
    (1+ (mlen (rest l)))))
```

```
(definec mlen (l :tl) :nat
  (if (endp l)
      (1+ (mlen (rest l)))
    0))
```

What if I wrote this?

DEMO

Questions?