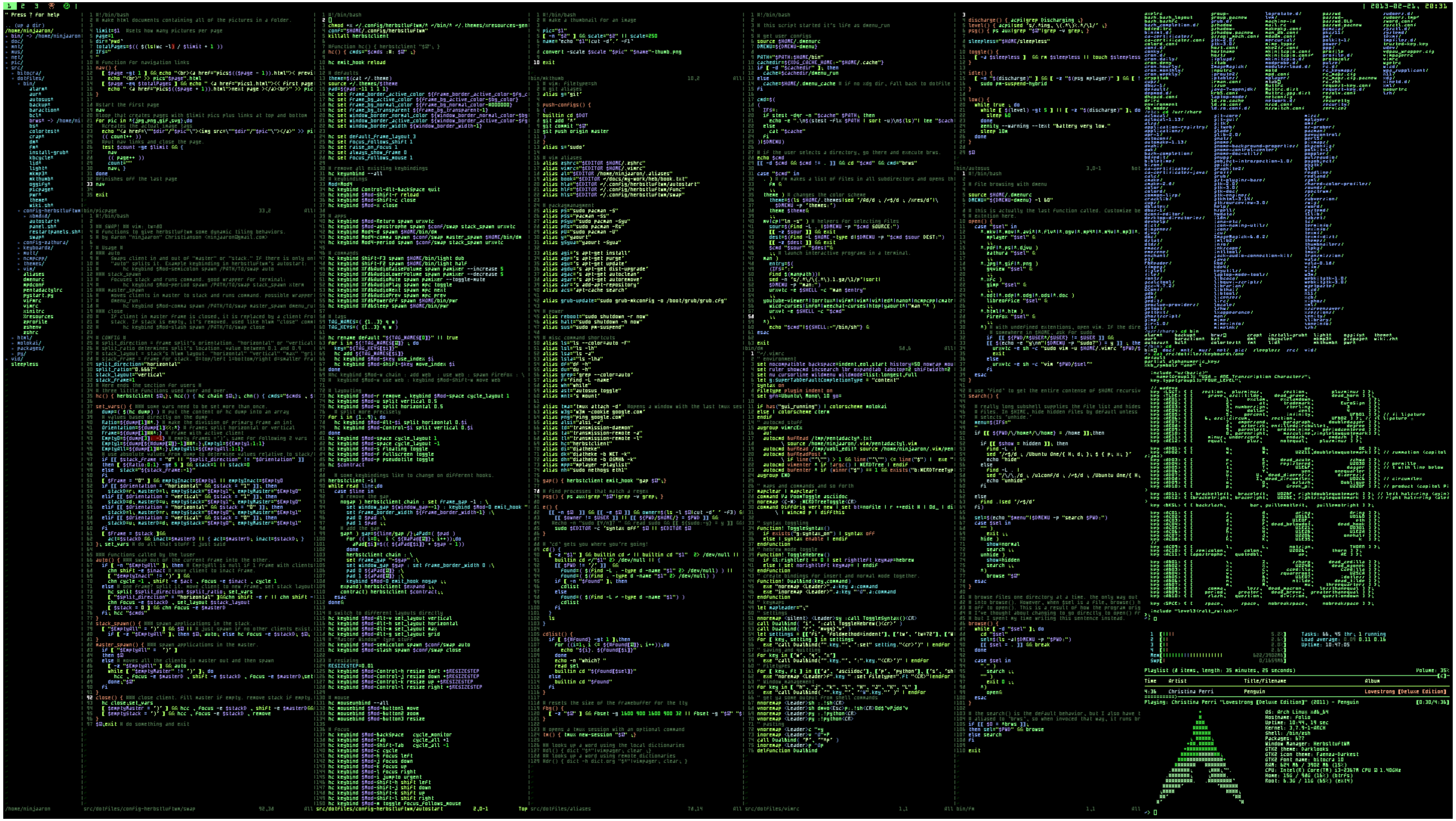
How Can We Write Large Programs Without Thinking?

Percy Liang

Stanford
ARTIFICIAL
INTELLIGENCE

Neural Abstract Machines & Program Induction Workshop

Dec. 10, 2016



```
| 2013-02-21, 20:31
```

```
  1  [||||                        5.2%]     Tasks: 66, 45 thr; 1 running
  2  [||                          2.6%]     Load average: 0.09 0.11 0.16
  3  [||                          2.0%]     Uptime: 10:47:05
  4  [||                          2.6%]
  Mem[|||||||||||||||||||     622/3902MB]
  Swp[|                         0/1659MB]
```

```
Playlist (8 items, length: 35 minutes, 25 seconds)                    Volume: 35%
──────────────────────────────────────────────────────────────────────────[c]─
Time    Artist              Title/Filename             Album
──────────────────────────────────────────────────────────────────────────────
4:36    Christina Perri     Penguin                    Lovestrong [Deluxe Edition]
==========>───────────────────────────────────────────────────────────────────
Playing: Christina Perri "Lovestrong [Deluxe Edition]" (2011) - Penguin  [0:30/4:36]
```

```
OS: Arch Linux x86_64
Hostname: Folio
Uptime: 10:44, 19 sec
Kernel: 3.7.9-1-ARCH
Shell: /bin/zsh
Packages: 677
Window Manager: HerbstluftWM
GTK2 theme: Darklooks
GTK2 icon theme: Faenza-Darkest
GTK2 font name: bitocra 10
RAM: 624 Mb / 3902 Mb (15%)
CPU: Intel(R) Core(TM) i3-2367M CPU @ 1.40GHz
Home: 15G / 98G (16%) (btrfs)
Root: 6.3G / 11G (65%) (ext4)
```

Linux: 15 million lines of code

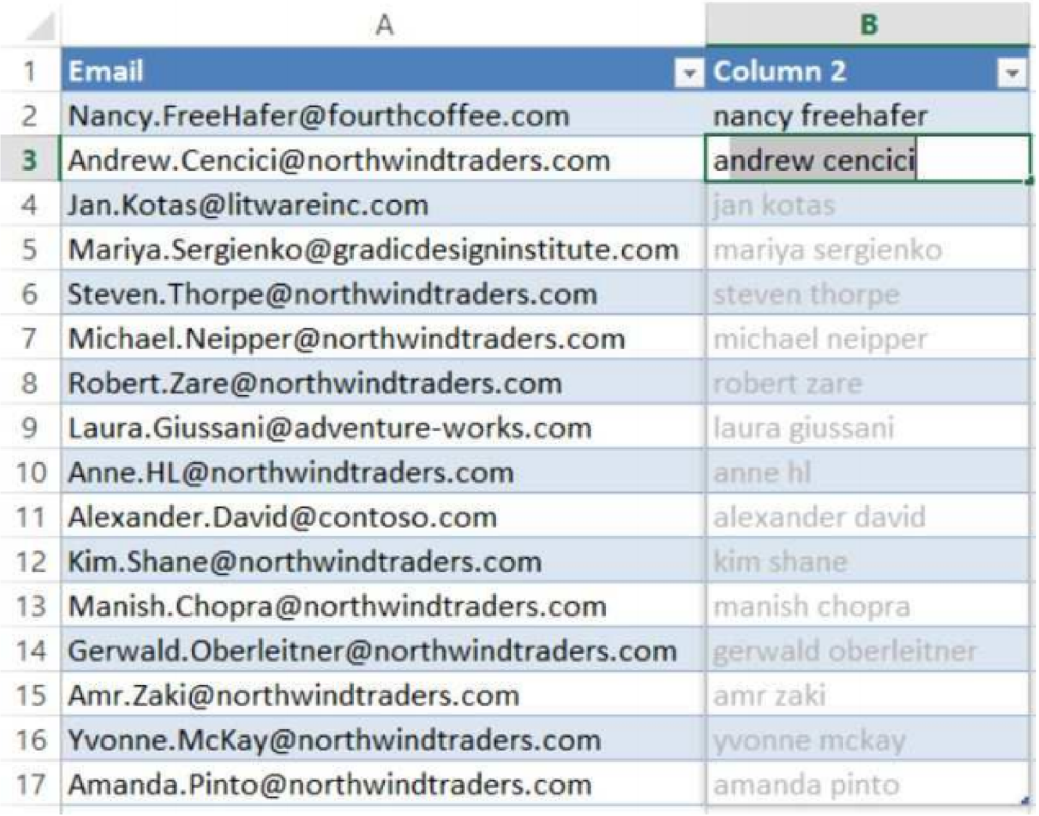
Windows 7: 40 million lines of code

Google: 2 billion lines of code

Where we are...

[Harris & Gulwani, 2011]

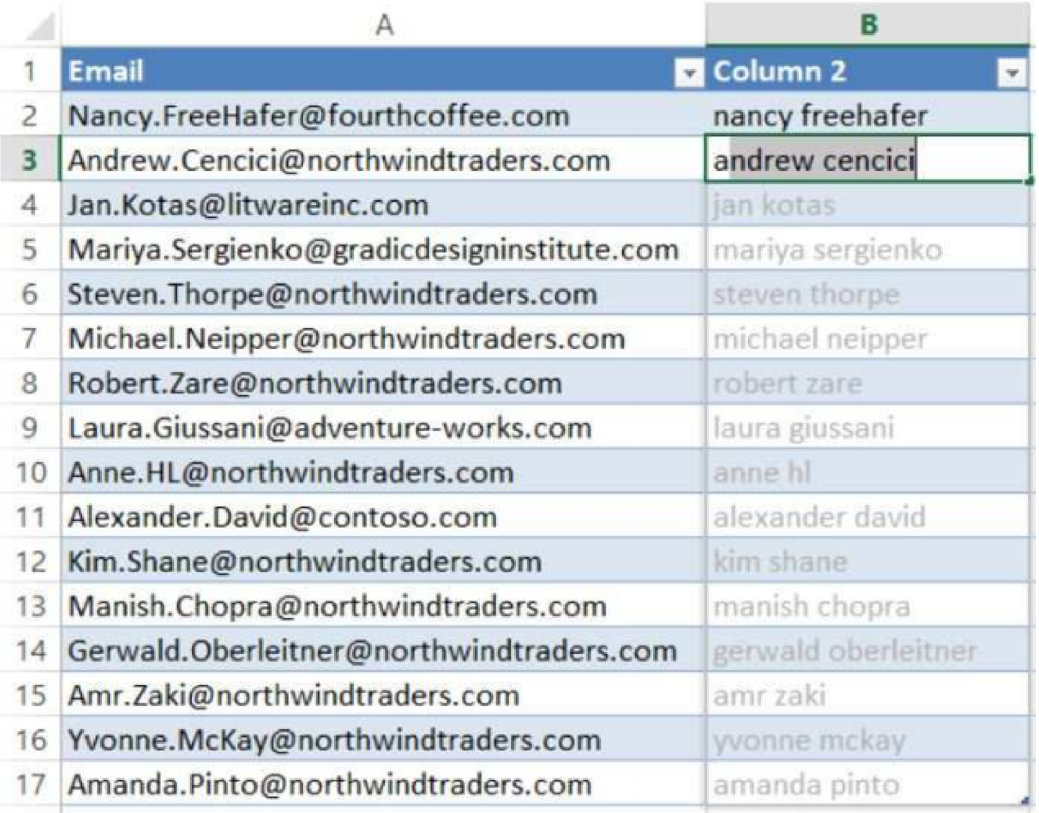
End-user programming



	A	B
1	Email	Column 2
2	Nancy.FreeHafer@fourthcoffee.com	nancy freehafer
3	Andrew.Cencici@northwindtraders.com	andrew cencici
4	Jan.Kotas@litwareinc.com	jan kotas
5	Mariya.Sergienko@gradicdesigninstitute.com	mariya sergienko
6	Steven.Thorpe@northwindtraders.com	steven thorpe
7	Michael.Neipper@northwindtraders.com	michael neipper
8	Robert.Zare@northwindtraders.com	robert zare
9	Laura.Giussani@adventure-works.com	laura giussani
10	Anne.HL@northwindtraders.com	anne hl
11	Alexander.David@contoso.com	alexander david
12	Kim.Shane@northwindtraders.com	kim shane
13	Manish.Chopra@northwindtraders.com	manish chopra
14	Gerwald.Oberleitner@northwindtraders.com	gerwald oberleitner
15	Amr.Zaki@northwindtraders.com	amr zaki
16	Yvonne.McKay@northwindtraders.com	yvonne mckay
17	Amanda.Pinto@northwindtraders.com	amanda pinto

[Harris & Gulwani, 2011]

End-user programming

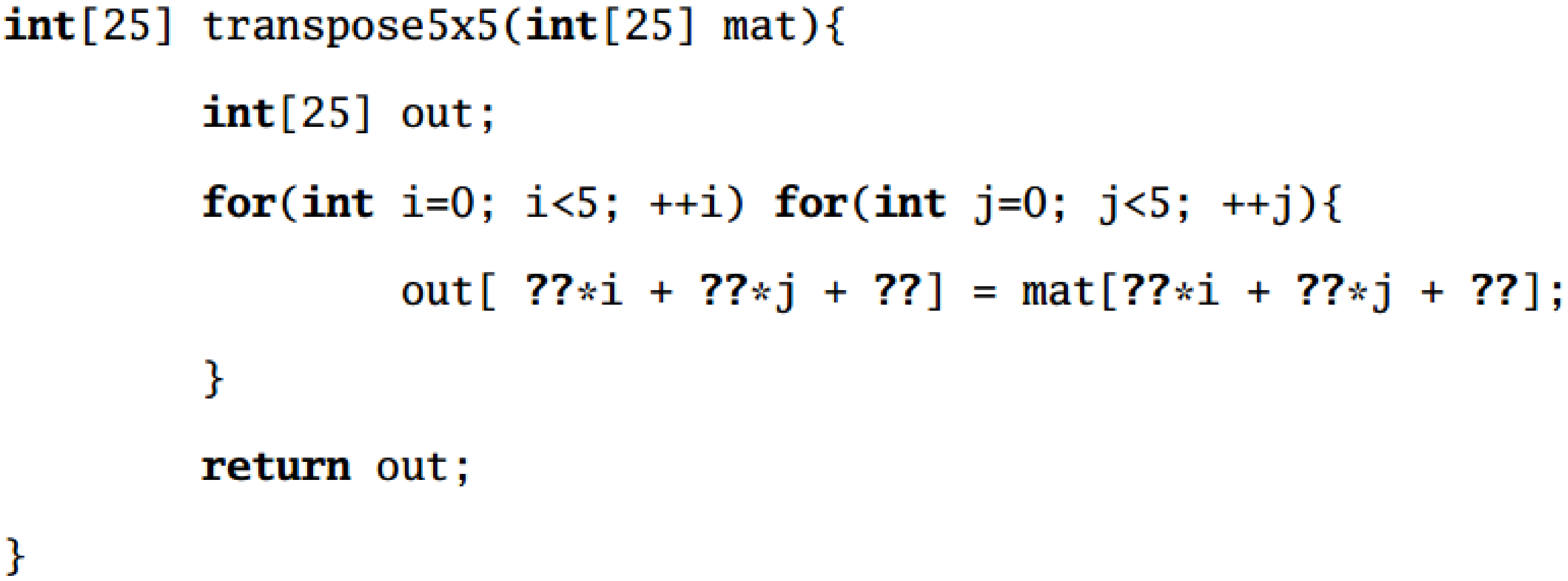


	A	B
1	Email	Column 2
2	Nancy.FreeHafer@fourthcoffee.com	nancy freehafer
3	Andrew.Cencici@northwindtraders.com	andrew cencici
4	Jan.Kotas@litwareinc.com	jan kotas
5	Mariya.Sergienko@gradicdesigninstitute.com	mariya sergienko
6	Steven.Thorpe@northwindtraders.com	steven thorpe
7	Michael.Neipper@northwindtraders.com	michael neipper
8	Robert.Zare@northwindtraders.com	robert zare
9	Laura.Giussani@adventure-works.com	laura giussani
10	Anne.HL@northwindtraders.com	anne hl
11	Alexander.David@contoso.com	alexander david
12	Kim.Shane@northwindtraders.com	kim shane
13	Manish.Chopra@northwindtraders.com	manish chopra
14	Gerwald.Oberleitner@northwindtraders.com	gerwald oberleitner
15	Amr.Zaki@northwindtraders.com	amr zaki
16	Yvonne.McKay@northwindtraders.com	yvonne mckay
17	Amanda.Pinto@northwindtraders.com	amanda pinto

Concatenate(ToLower(Substring(v,WordToken,1)), ” ”,
ToLower(Substring(v,WordToken,2)))

[Solar-Lezama, 2008]

Sketching



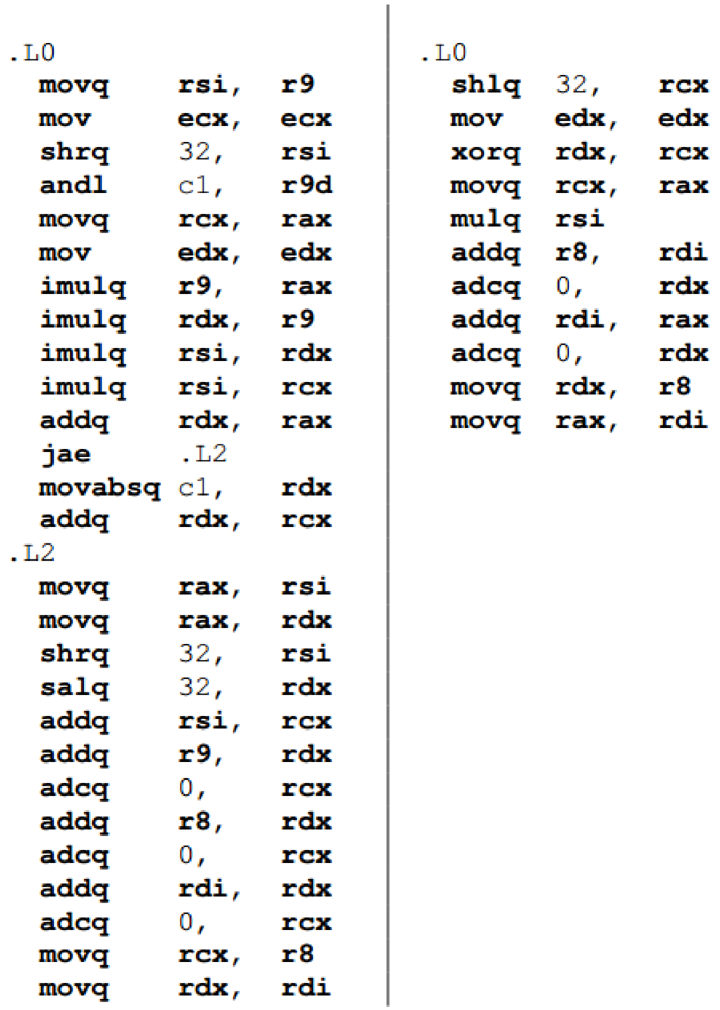
```
int[25] transpose5x5(int[25] mat){
        int[25] out;
        for(int i=0; i<5; ++i) for(int j=0; j<5; ++j){
                out[ ??*i + ??*j + ??] = mat[??*i + ??*j + ??];
        }
        return out;
}
```

Set up the skeleton, only fill in ??

[Schkufza/Sharma/Aiken, 2013]

Stochastic superoptimization (STOKE)

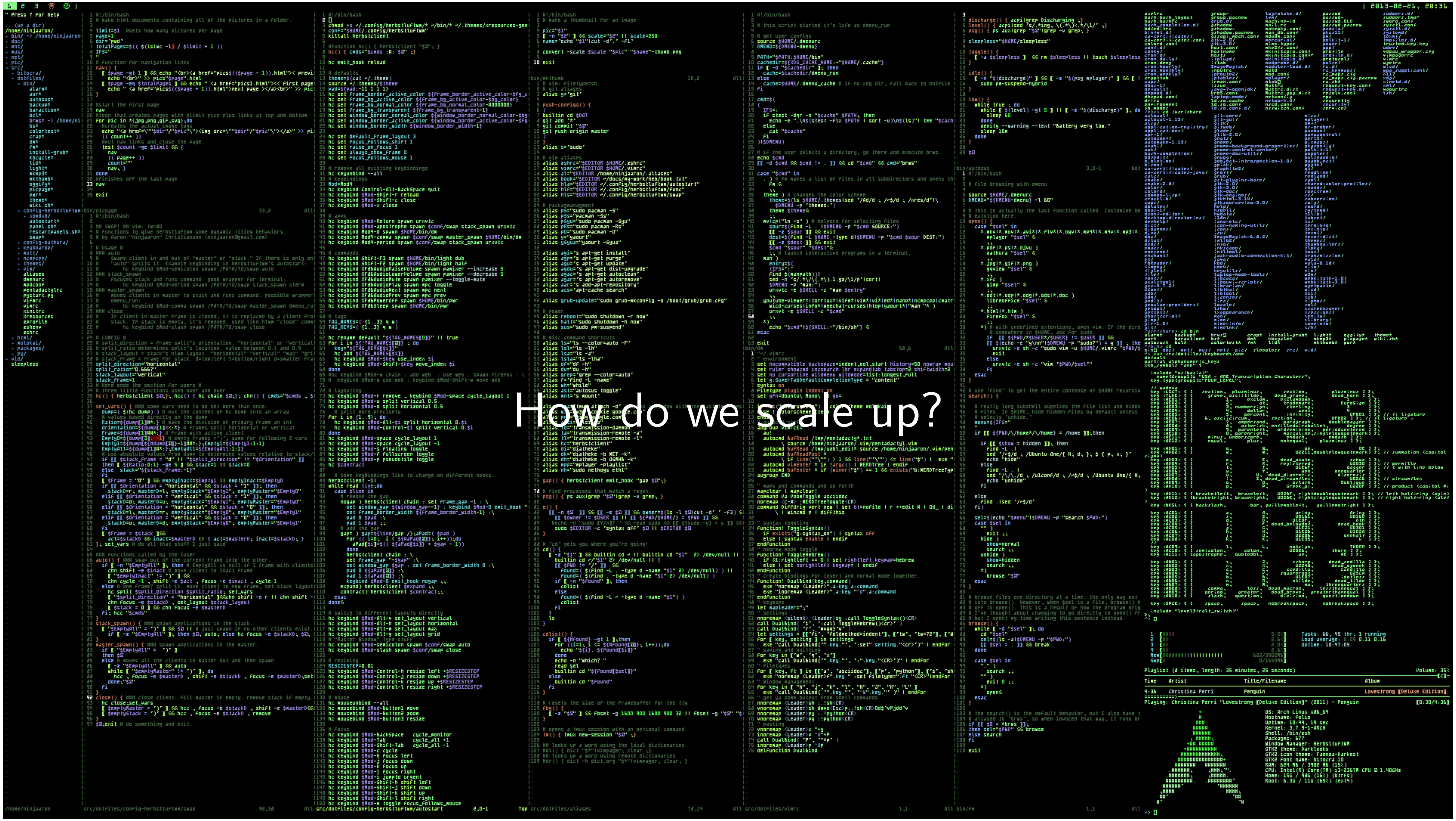
Montgomery multiplication:



```
.L0
  movq    rsi,  r9
  mov     ecx,  ecx
  shrq    32,   rsi
  andl    c1,   r9d
  movq    rcx,  rax
  mov     edx,  edx
  imulq   r9,   rax
  imulq   rdx,  r9
  imulq   rsi,  rdx
  imulq   rsi,  rcx
  addq    rdx,  rax
  jae     .L2
  movabsq c1,   rdx
  addq    rdx,  rcx
.L2
  movq    rax,  rsi
  movq    rax,  rdx
  shrq    32,   rsi
  salq    32,   rdx
  addq    rsi,  rcx
  addq    r9,   rdx
  adcq    0,    rcx
  addq    r8,   rdx
  adcq    0,    rcx
  addq    rdi,  rdx
  adcq    0,    rcx
  movq    rcx,  r8
  movq    rdx,  rdi
```

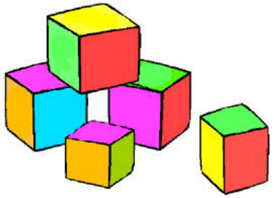
```
.L0
  shlq  32,   rcx
  mov   edx,  edx
  xorq  rdx,  rcx
  movq  rcx,  rax
  mulq  rsi
  addq  r8,   rdi
  adcq  0,    rdx
  addq  rdi,  rax
  adcq  0,    rdx
  movq  rdx,  r8
  movq  rax,  rdi
```

STOKE code 16 lines shorter, 1.6x faster than gcc



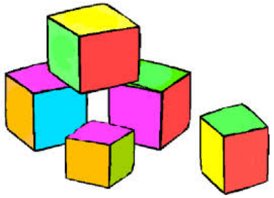
How do we scale up?

Modularity (model)

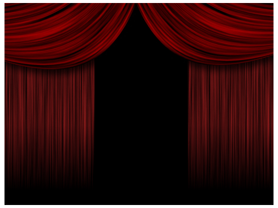


Natural language specifications

Moduarity (search)



Final remarks



[Liang et al., 2010, ICML]

A property of programs

```
def min(x, y): return x if x ¡ y else y
```

[Liang et al., 2010, ICML]

A property of programs

def min(x, y): return x if x ¡ y else y

def max(x, y): return x if x ¿ y else y

[Liang et al., 2010, ICML]

A property of programs

```
def min(x, y): return x if x ¡ y else y

def max(x, y): return x if x ¿ y else y
```

Programs share common subprograms

[Liang et al., 2010, ICML]

A property of programs

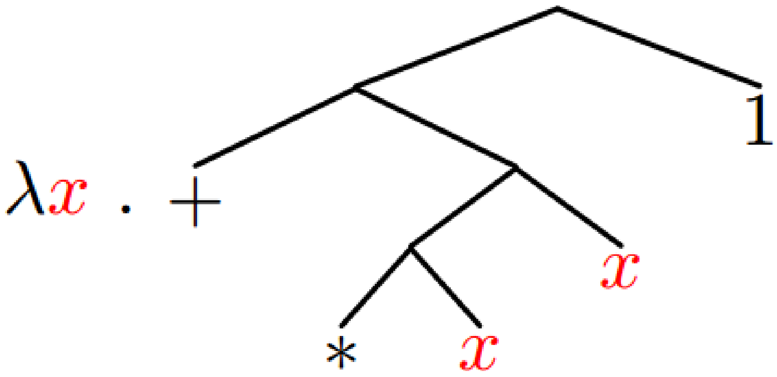
```
def min(x, y): return x if x ¡ y else y

def max(x, y): return x if x ¿ y else y
```

Programs share common subprograms

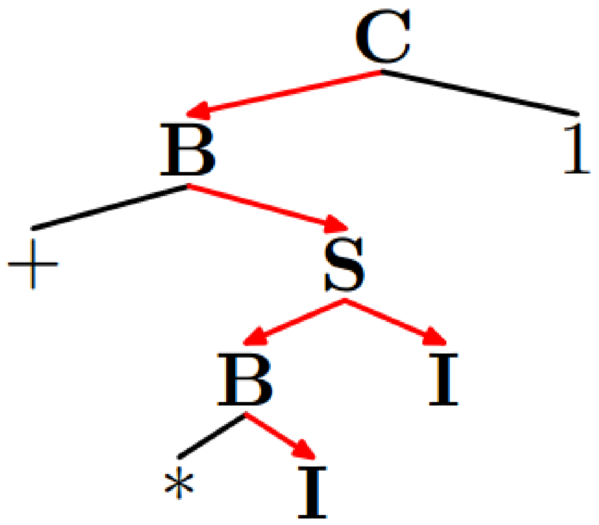
Problems:

- Part that changes embedded deeply in program
- Variables make it hard to extract subprograms

A new representation of programs

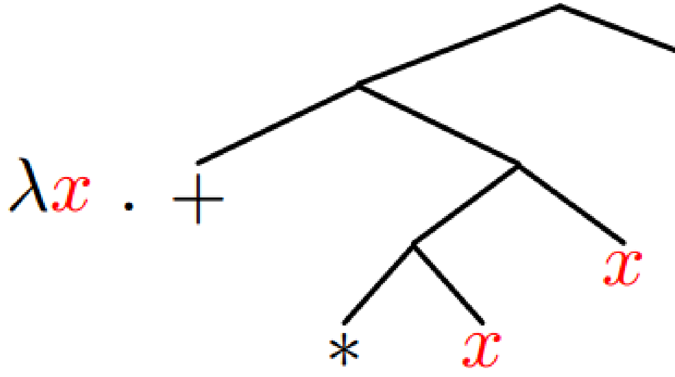


Lambda calculus

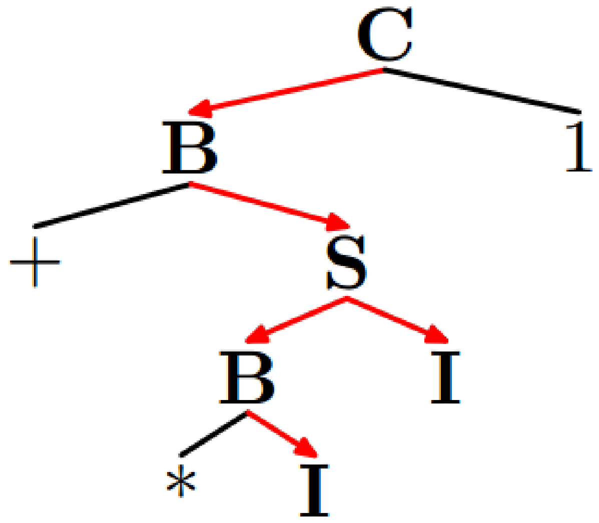


Combinatory logic

A new representation of programs



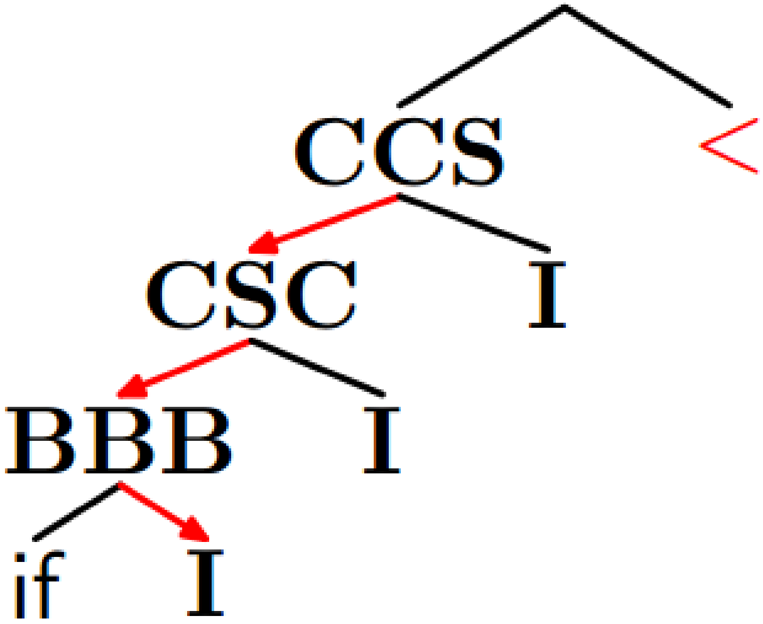
Lambda calculus

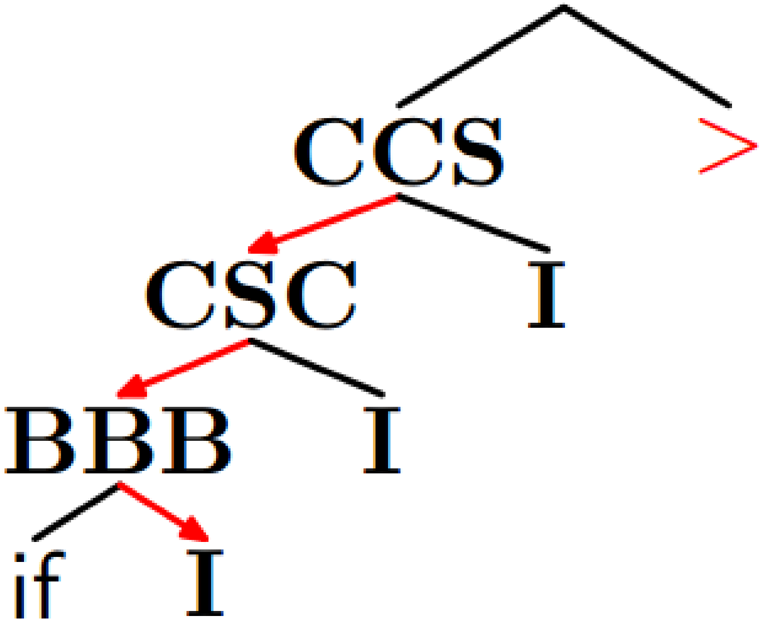


Combinatory logic

- Combinators B, C, S, I route encode how arguments get routed down
- Now subproblems are subtrees
- Extension of classic combinatory logic (Schönfinkel, 1924)

Representation of min and max





Adaptor grammars [Johnson, 2007]

```
$C_t \leftarrow []$ for each type $t$ [cached list of combinators]
  (notation: return* $c$ adds $c$ to $C_t$ and returns $c$)

$\textsc{GenCache}(t)$: [returns a combinator of type $t$]
  With probability $\frac{\alpha_0+N_t d}{\alpha_0+|C_t|}$:
    With probability $\lambda_0$:
      Return* a random primitive combinator (e.g., $+$, $3$, $\mathbf{I}$)
    Else:
      Choose a type $s$
      $x \leftarrow \textsc{GenCache}(s \to t)$
      $y \leftarrow \textsc{GenCache}(s)$
      Return* $(x, y)$
  Else:
    Return* $z \in C_t$ with probability $\frac{M_z-d}{|C_t|-N_t d}$
```

Inference: MCMC

Experimental results

24 text editing tasks [Lau, 2003]

Cardinals 5, Pirates 2

$\Downarrow$

GameScore[winner 'Cardinals'; loser 'Pirates'; scores [5, 2]]

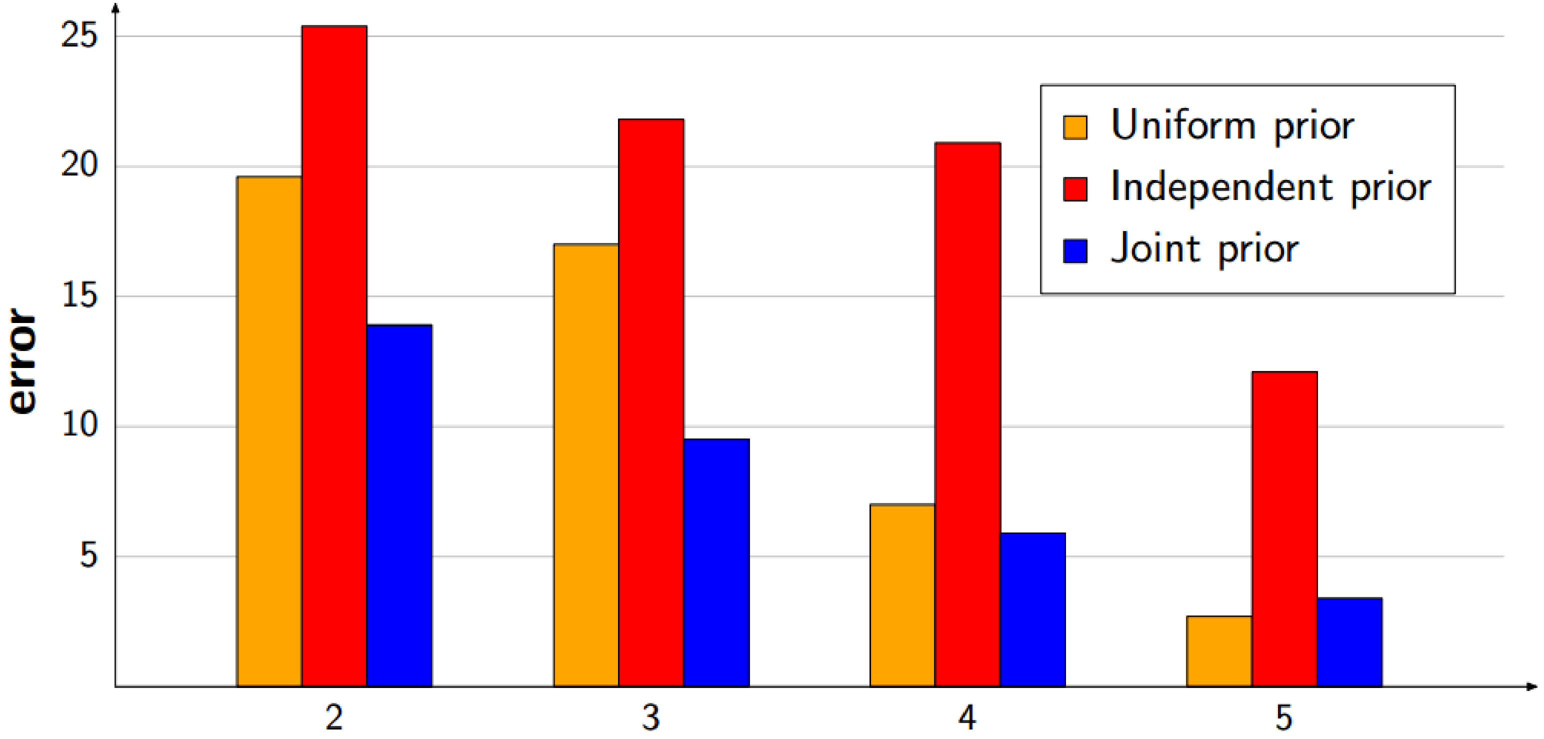
Experimental results

24 text editing tasks [Lau, 2003]

Cardinals 5, Pirates 2

⇓

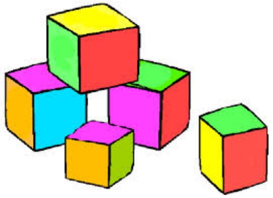
GameScore[winner 'Cardinals'; loser 'Pirates'; scores [5, 2]]



Summary so far

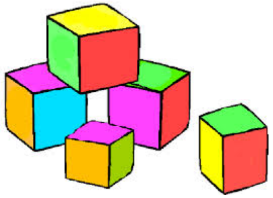
- Multi-task learning: if induce one program, should be easier to induce a related one
- Need to expose shared subprograms
- Use adaptor grammar over combinatory logic
- Cache becomes a library of useful primitives

Modularity (model)

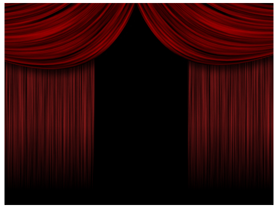


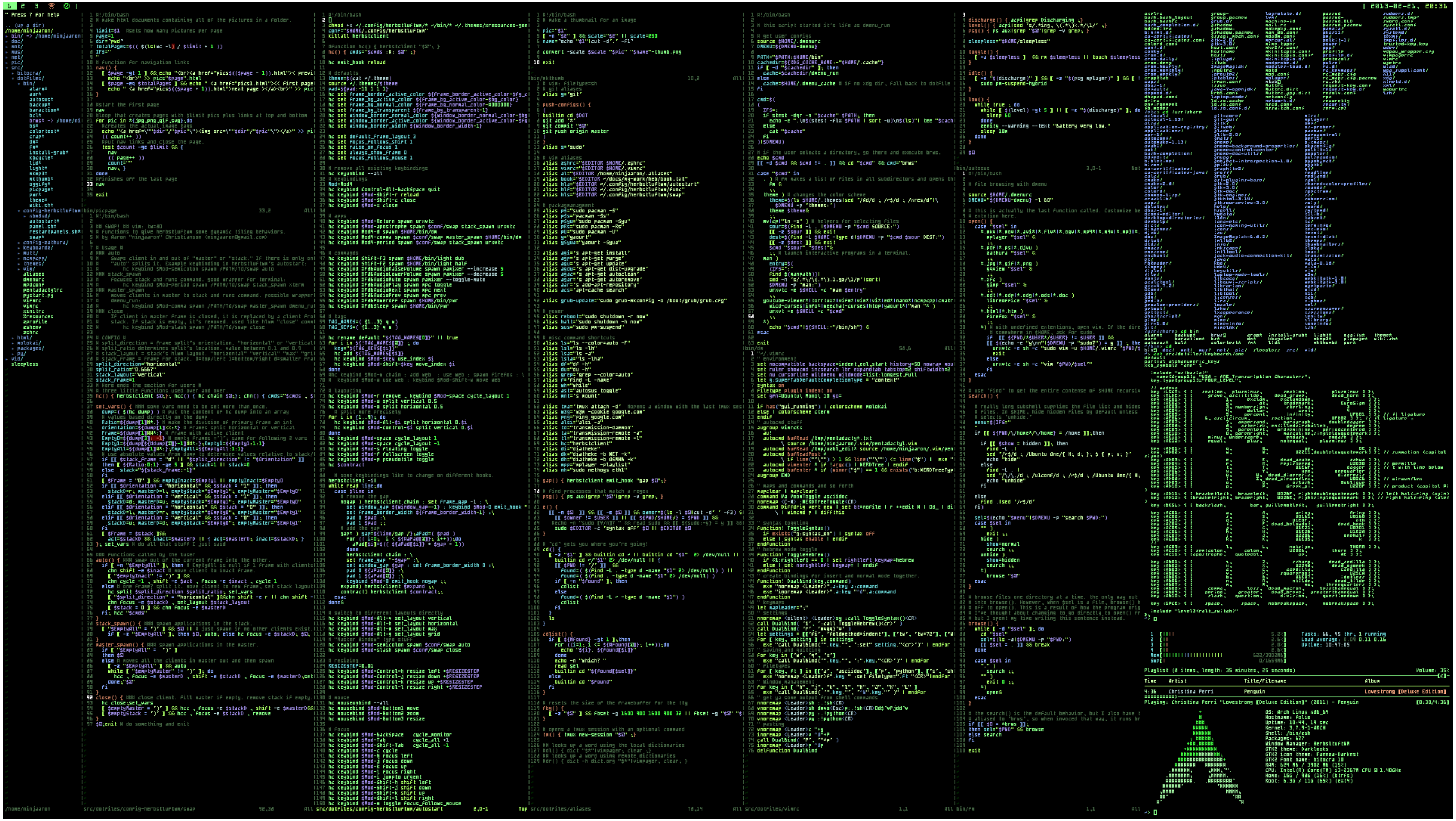
Natural language specifications

Moduarity (search)



Final remarks

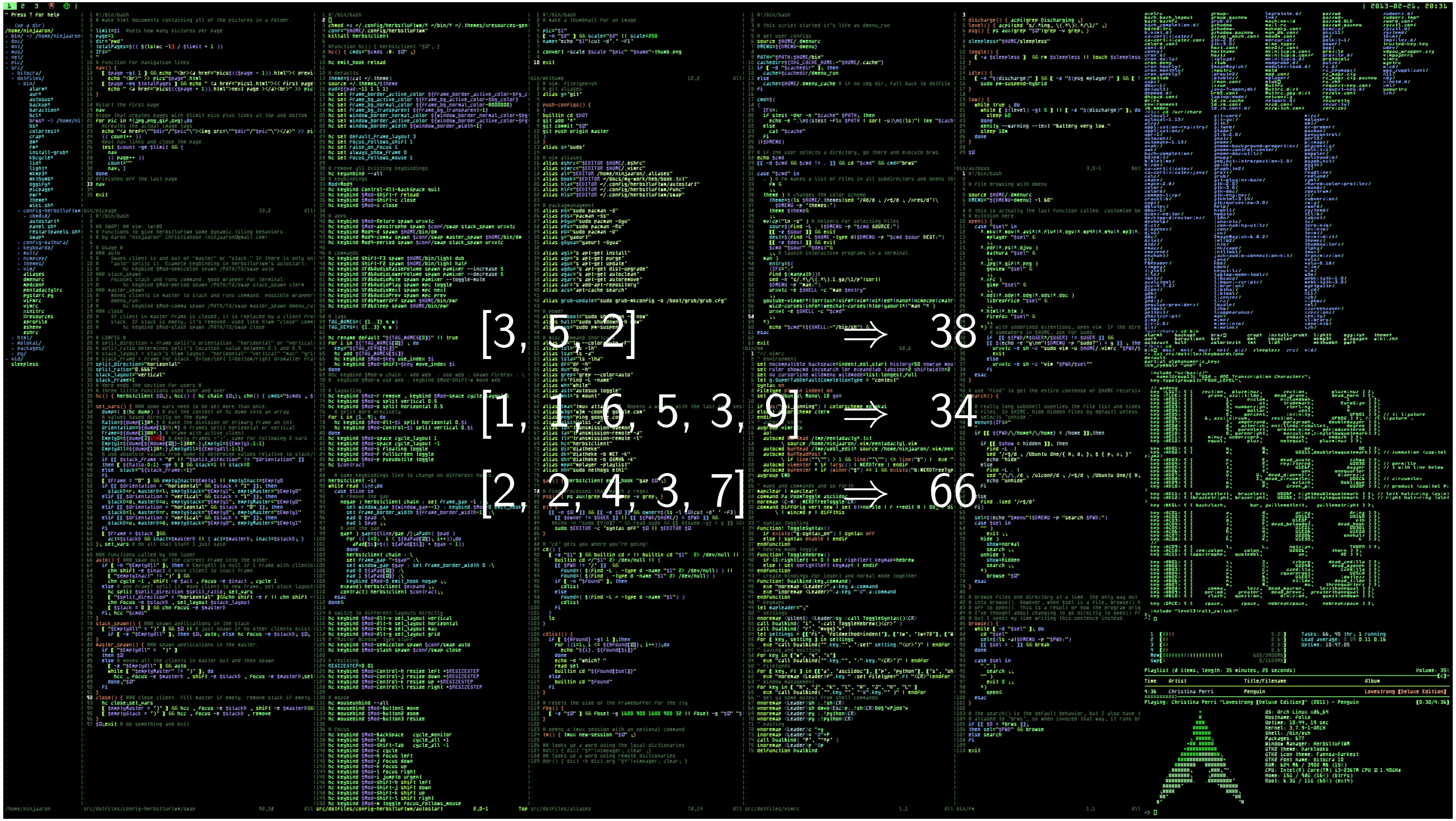




```
1  2  3                                                            | 2013-02-21, 20:31

" Press ? for help
.. (up a dir)
/home/ninjaaron/
▸ bin/ -> /home/ninjaaron
▸ doc/
▸ mnt/
▸ mus/
▸ net/
▸ pic/
▾ src/
  ▸ bitocra/
  ▾ dotfiles/
    ▾ bin/
        alarm*
        aur*
        autosus*
        backup*
        baraction*
        bcl*
        brws* -> /home/ni
        bs*
        colortest*
        crap*
        dm*
```

```
#!/bin/bash
# make html documents containing all of the pictures in a folder.

limit=$1  #sets how many pictures per page
page=1
dir=`pwd`
totalPages=$(( $(ls|wc -l) / $limit + 1 ))
IFS="
"
# function for navigation links
nav() {
  [ $page -gt 1 ] && echo "<br><a href="pics$(($page - 1)).html">< previ
    echo "<br>" >> pics"$page".html
  [ $page -eq $totalPages ] && echo " <a href="pics1.html"><< first page
    echo " <a href="pics$(($page + 1)).html">next page ></a><br>" >> pic
}

#start the first page
nav
#loop that creates pages with $limit pics plus links at top and bottom
for pic in *{jpg,png,gif,svg};do
  #creates the actual image tags
  echo "<a href=\""$dir"/"$pic"\"><img src=\""$dir"/"$pic"\"></a>" >> pi
  (( count++ ))
  #put nav links and close the page.
```

```
  include "level3(ralt_switch)"
};
~> □

  1  [||||                        5.2%]     Tasks: 66, 45 thr; 1 running
  2  [||                          2.6%]     Load average: 0.09 0.11 0.16
  3  [||                          2.0%]     Uptime: 10:47:05
  4  [||                          2.6%]
  Mem[||||||||||||||||||||  622/3902MB]
  Swp[|                        0/1659MB]

Playlist (8 items, length: 35 minutes, 25 seconds)                      Volume: 35%
Time    Artist              Title/Filename                 Album
4:36    Christina Perri     Penguin                        Lovestrong [Deluxe Edition]
==========>
Playing: Christina Perri "Lovestrong [Deluxe Edition]" (2011) - Penguin    [0:30/4:36]

                     OS: Arch Linux x86_64
                     Hostname: folio
                     Uptime: 10:44, 19 sec
                     Kernel: 3.7.9-1-ARCH
                     Shell: /bin/zsh
                     Packages: 677
                     Window Manager: HerbstluftWM
                     GTK2 theme: Darklooks
                     GTK2 icon theme: Faenza-Darkest
                     GTK2 font name: bitocra 10
                     RAM: 624 Mb / 3902 Mb (15%)
                     CPU: Intel(R) Core(TM) i3-2367M CPU @ 1.40GHz
                     Home: 15G / 98G (16%) (btrfs)
                     Root: 6.3G / 11G (65%) (ext4)
~> □
```



$[3, 5, 2] \Rightarrow 38$

$[1, 1, 6, 5, 3, 9] \Rightarrow 34$

$[2, 2, 4, 3, 7] \Rightarrow 66$

sum of squares of primes in the list

People code from partial specifications, not from examples!

[database]

Language to programs

What is the largest city in Europe by population?

[database]

Language to programs

What is the largest city in Europe by population?

semantic parsing

Cities

[database]

Language to programs

What is the largest city in Europe by population?

semantic parsing

Cities Europe

[database]

Language to programs

What is the largest city in Europe by population?

semantic parsing

Cities ContainedBy(Europe)

Language to programs

[database]

What is the largest city in Europe by population?

semantic parsing

$\mathrm{Cities} \cap \mathrm{ContainedBy}(\mathrm{Europe})$

[database]

Language to programs

What is the largest city in Europe by population?

↓ semantic parsing

$\text{Cities} \cap \text{ContainedBy}(\text{Europe})$ Population

[database]

Language to programs

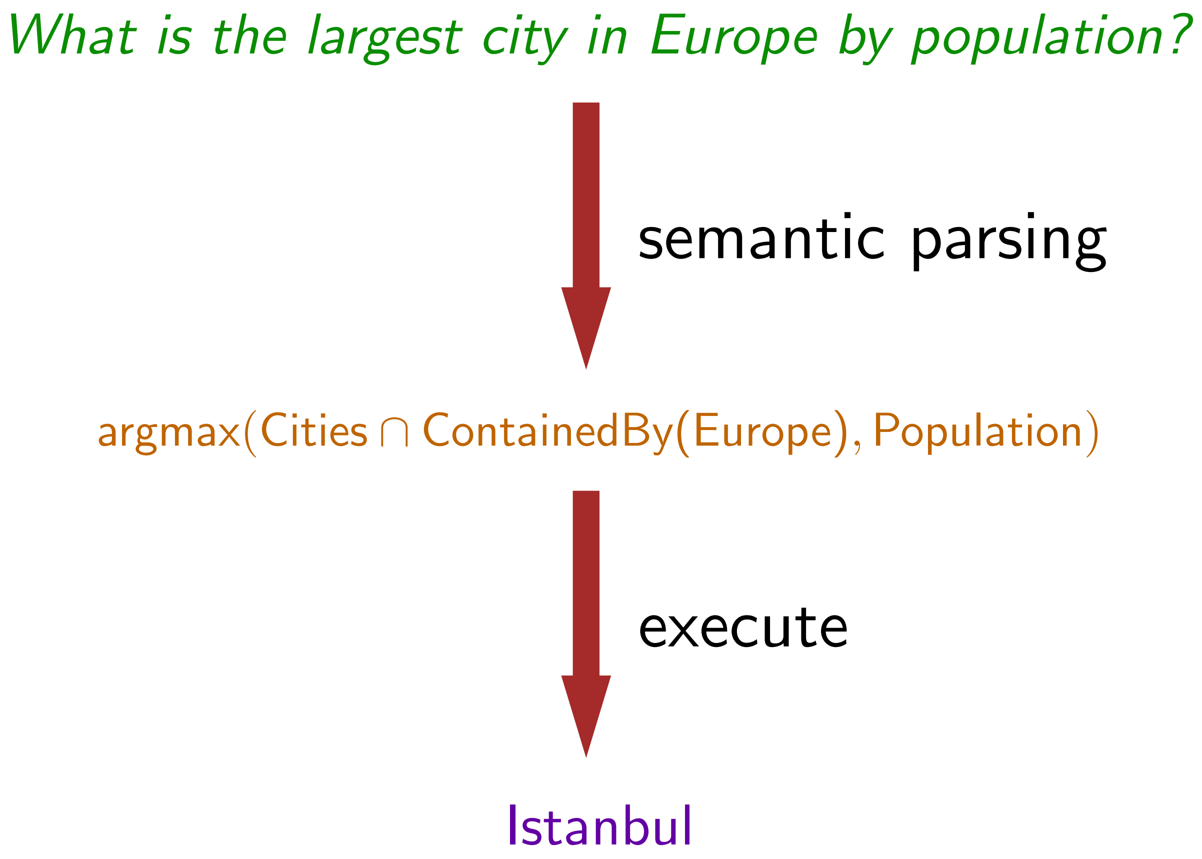
What is the largest city in Europe by population?

semantic parsing

$\text{argmax}(\text{Cities} \cap \text{ContainedBy}(\text{Europe}), \text{Population})$

[database]

Language to programs



What is the largest city in Europe by population?

semantic parsing

$\text{argmax}(\text{Cities} \cap \text{ContainedBy}(\text{Europe}), \text{Population})$

execute

Istanbul

Language to programs

Remind me to buy milk after my last meeting on Monday.

[calendar]

Language to programs

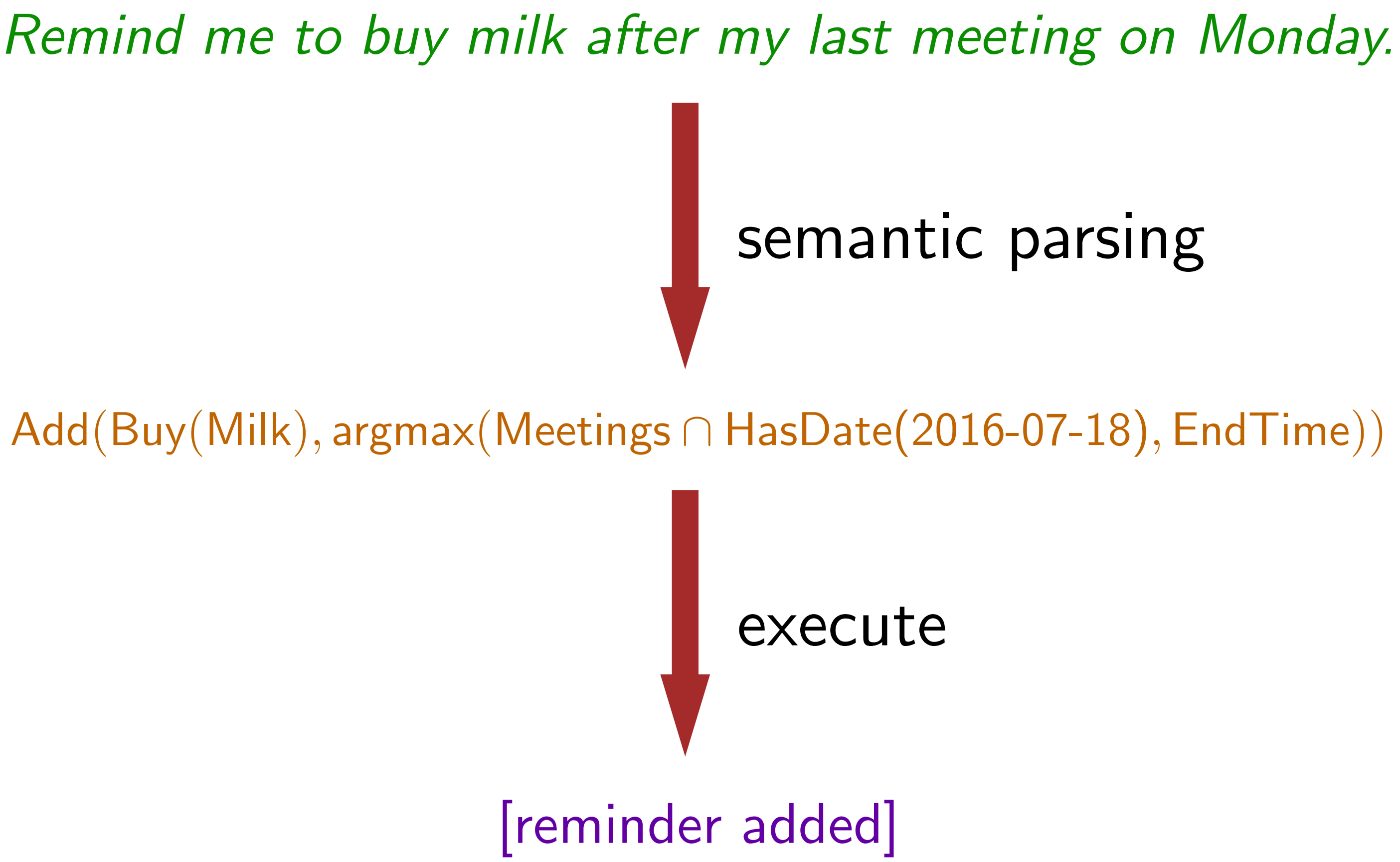
Remind me to buy milk after my last meeting on Monday.

semantic parsing

$\text{Add}(\text{Buy}(\text{Milk}), \text{argmax}(\text{Meetings} \cap \text{HasDate}(2016\text{-}07\text{-}18), \text{EndTime}))$

Language to programs

[calendar]



Remind me to buy milk after my last meeting on Monday.

semantic parsing

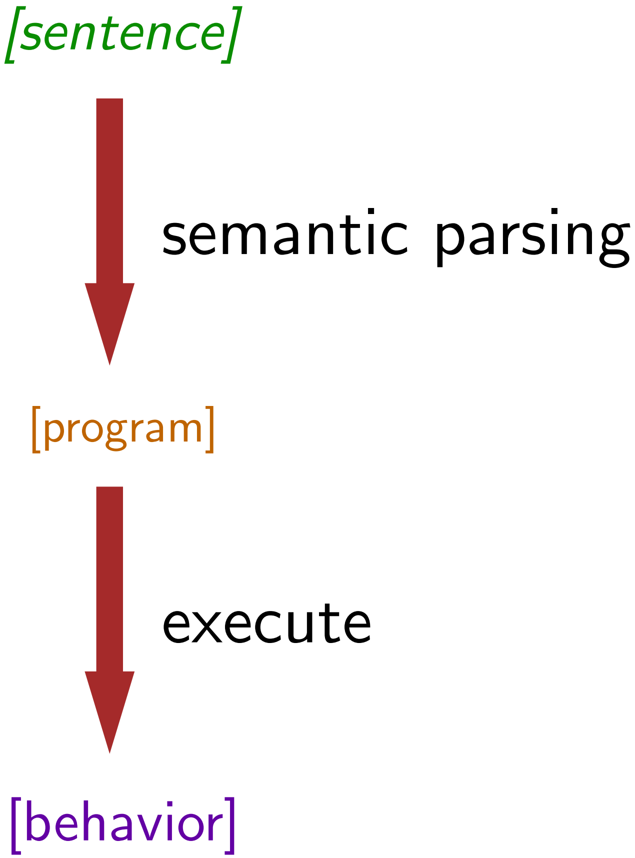
$\mathrm{Add}(\mathrm{Buy}(\mathrm{Milk}), \mathrm{argmax}(\mathrm{Meetings} \cap \mathrm{HasDate}(2016\text{-}07\text{-}18), \mathrm{EndTime}))$

execute

[reminder added]

[context]

Language to programs



[sentence]

semantic parsing

[program]

execute

[behavior]

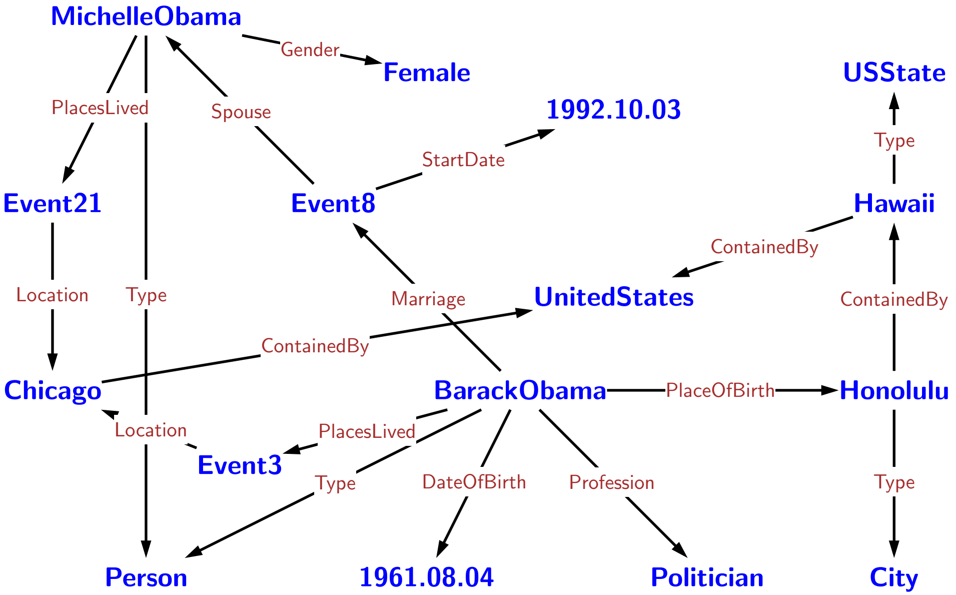
A brief history of semantic parsing

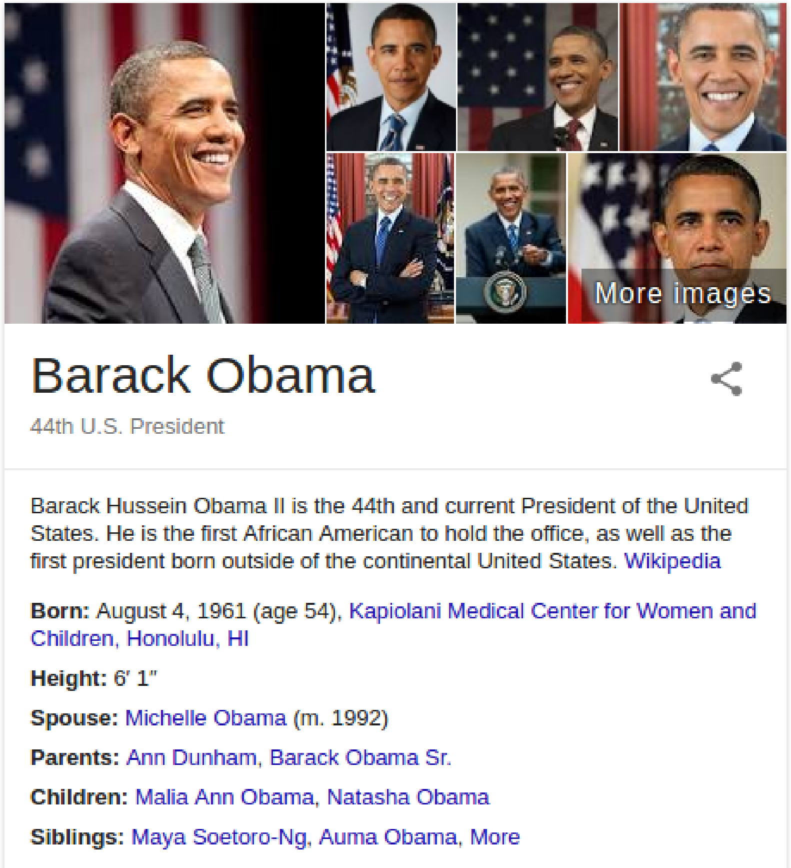
GeoQuery [Zelle & Mooney 1996]	
Inductive logic programming [Tang & Mooney 2001]	
	CCG [Zettlemoyer & Collins 2005]
String kernels [Kate & Mooney 2006]	
Synchronous grammars [Wong & Mooney 2007]	Relaxed CCG [Zettlemoyer & Collins 2007]
	Learning from world [Clarke et al. 2010]
Higher-order unification [Kwiatkowski et al. 2011]	Learning from answers [Liang et al. 2011]
Language + vision [Matsusek et al. 2012]	
	Regular expressions [Kushman et al. 2013]
Large-scale KBs [Berant et al.; Kwiatkowski et al. 2013]	
	Instruction following [Artzi & Zettlemoyer 2013]
Reduction to paraphrasing [Berant & Liang 2014]	
Compositionality on tables [Pasupat & Liang, 2015]	Dataset from logical forms [Wang et al. 2015]

[Bollacker, 2008; Google, 2013]

QA on Freebase

100M entities (nodes)   1B assertions (edges)





More images

Barack Obama

44th U.S. President

Barack Hussein Obama II is the 44th and current President of the United States. He is the first African American to hold the office, as well as the first president born outside of the continental United States. Wikipedia

Born: August 4, 1961 (age 54), Kapiolani Medical Center for Women and Children, Honolulu, HI

Height: 6′ 1″

Spouse: Michelle Obama (m. 1992)

Parents: Ann Dunham, Barack Obama Sr.

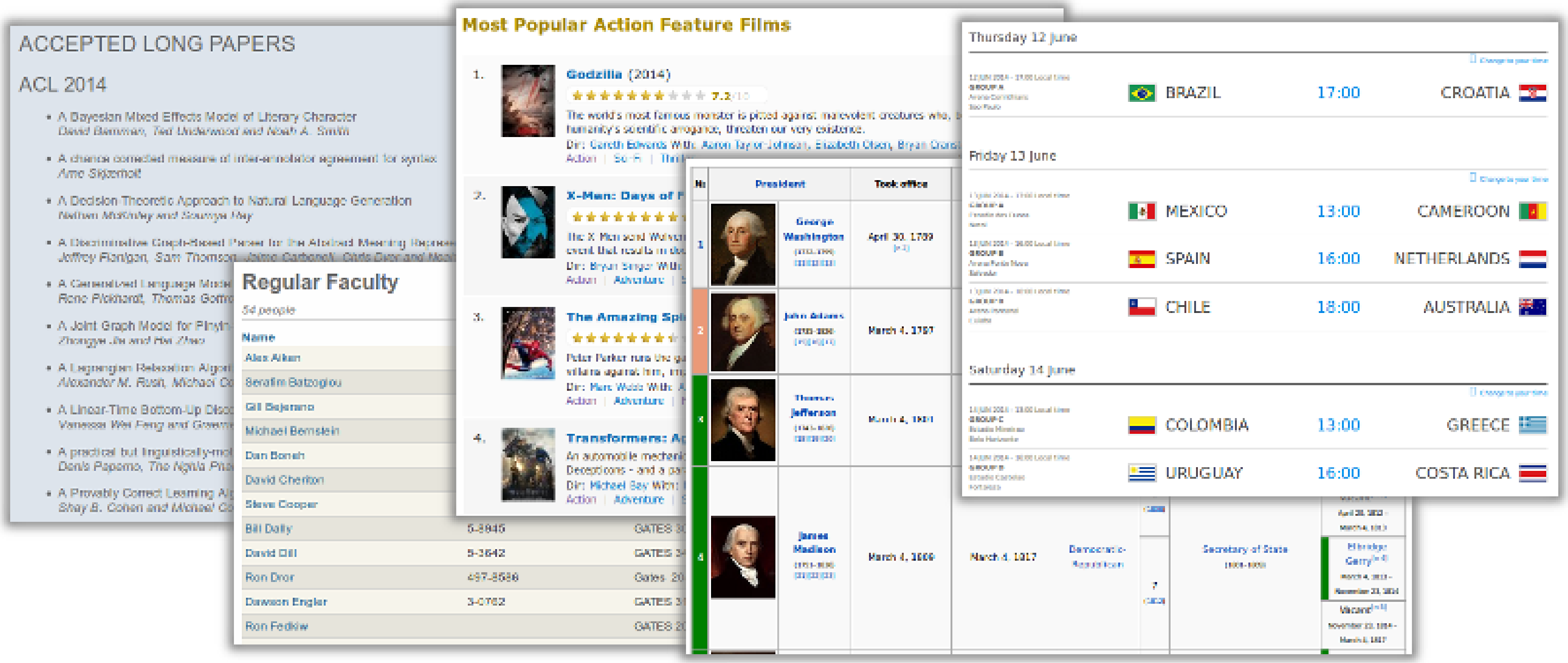
Children: Malia Ann Obama, Natasha Obama

Siblings: Maya Soetoro-Ng, Auma Obama, More

WebQuestions [Berant et al., 2013]

QA on semi-structured data

154 million tables on the web [Cafarella et al. 2008]



ACCEPTED LONG PAPERS

ACL 2014

- A Bayesian Mixed Effects Model of Literary Character
  David Bamman, Ted Underwood and Noah A. Smith
- A chance corrected measure of inter-annotator agreement for syntax
  Arne Skjærholt
- A Decision-Theoretic Approach to Natural Language Generation
  Nathan McKinley and Soumya Ray
- A Discriminative Graph-Based Parser for the Abstract Meaning Represe
  Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer and Noah
- A Generalized Language Mode
  Rene Pickhardt, Thomas Gottro
- A Joint Graph Model for Pinyin
  Zhongye Jia and Hai Zhao
- A Lagrangian Relaxation Algori
  Alexander M. Rush, Michael Co
- A Linear-Time Bottom-Up Disco
  Vanessa Wei Feng and Graeme
- A practical but linguistically-mo
  Denis Paperno, The Nghia Pha
- A Provably Correct Learning Al
  Shay B. Cohen and Michael Co

Regular Faculty

54 people

Name		
Alex Aiken		
Serafim Batzoglou		
Gill Bejerano		
Michael Bernstein		
Dan Boneh		
David Cheriton		
Steve Cooper		
Bill Dally	5-3845	GATES 3
David Dill	5-3642	GATES 3
Ron Dror	497-8586	Gates 20
Dawson Engler	3-0762	GATES 3
Ron Fedkiw		GATES 2

Most Popular Action Feature Films

1. **Godzilla** (2014)
   ★★★★★★★☆☆☆ 7.2/10
   The world's most famous monster is pitted against malevolent creatures who, b humanity's scientific arrogance, threaten our very existence.
   Dir: Gareth Edwards With: Aaron Taylor-Johnson, Elizabeth Olsen, Bryan Crans
   Action | Sci-Fi | Thri
2. **X-Men: Days of F**
   ★★★★★★★★
   The X-Men send Wolver event that results in do
   Dir: Bryan Singer With:
   Action | Adventure | S
3. **The Amazing Spi**
   ★★★★★★★★
   Peter Parker runs the ga villains against him, imp
   Dir: Marc Webb With: A
   Action | Adventure | F
4. **Transformers: Ag**
   An automobile mechanic Decepticons - and a pa
   Dir: Michael Bay With:
   Action | Adventure | S

No.		President	Took office					
1		George Washington	April 30, 1789					
2		John Adams	March 4, 1797					
3		Thomas Jefferson	March 4, 1801					
4		James Madison	March 4, 1809	March 4, 1817	Democratic-Republican	Secretary of State	Elbridge Gerry	Vacant

Thursday 12 June

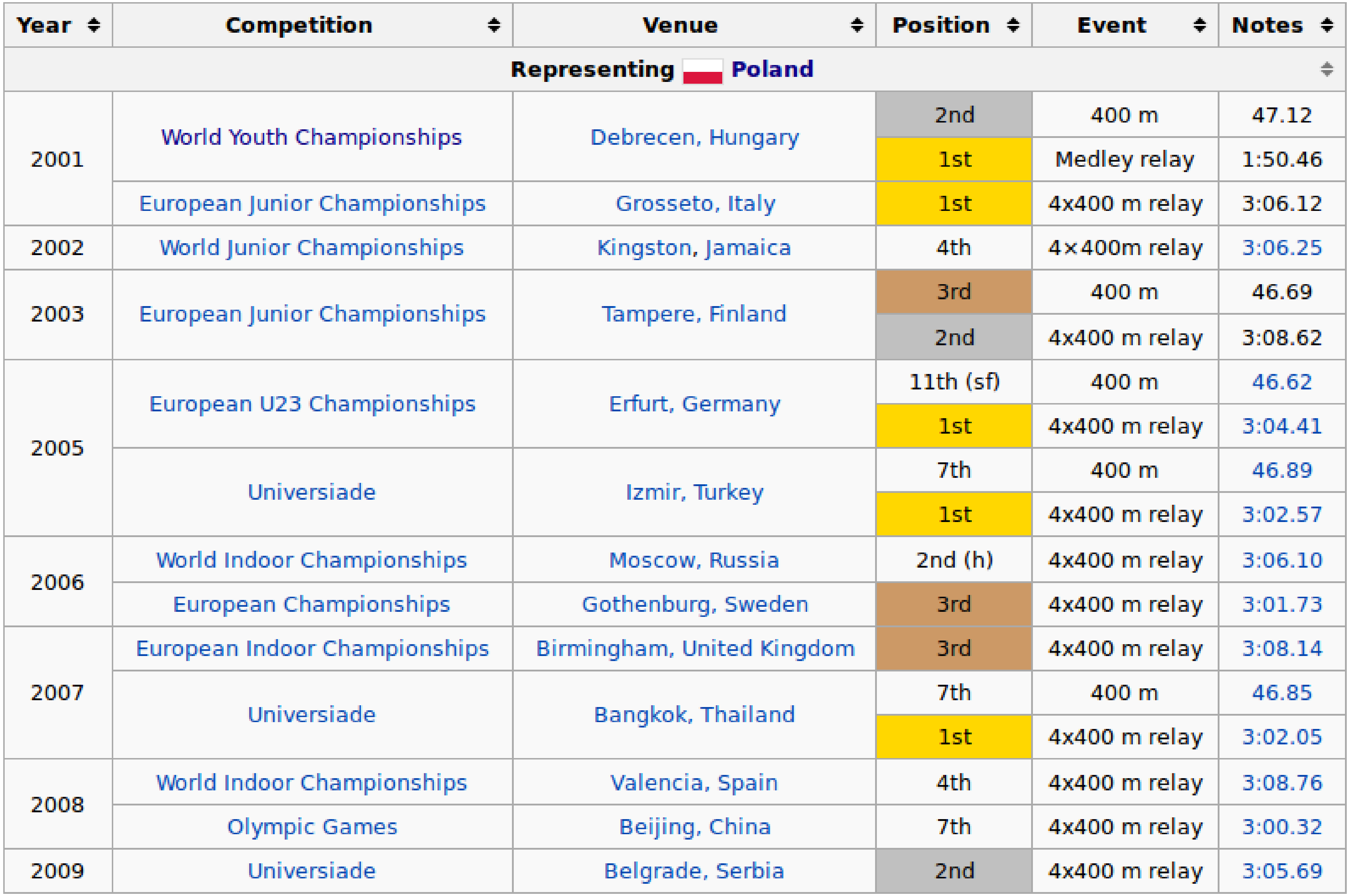
GROUP A	BRAZIL	17:00	CROATIA	

Friday 13 June

	MEXICO	13:00	CAMEROON	
GROUP B	SPAIN	16:00	NETHERLANDS	
	CHILE	18:00	AUSTRALIA	

Saturday 14 June

GROUP C	COLOMBIA	13:00	GREECE	
GROUP D	URUGUAY	16:00	COSTA RICA	



Year	Competition	Venue	Position	Event	Notes
Representing Poland					
2001	World Youth Championships	Debrecen, Hungary	2nd	400 m	47.12
			1st	Medley relay	1:50.46
	European Junior Championships	Grosseto, Italy	1st	4x400 m relay	3:06.12
2002	World Junior Championships	Kingston, Jamaica	4th	4×400m relay	3:06.25
2003	European Junior Championships	Tampere, Finland	3rd	400 m	46.69
			2nd	4x400 m relay	3:08.62
2005	European U23 Championships	Erfurt, Germany	11th (sf)	400 m	46.62
			1st	4x400 m relay	3:04.41
	Universiade	Izmir, Turkey	7th	400 m	46.89
			1st	4x400 m relay	3:02.57
2006	World Indoor Championships	Moscow, Russia	2nd (h)	4x400 m relay	3:06.10
	European Championships	Gothenburg, Sweden	3rd	4x400 m relay	3:01.73
2007	European Indoor Championships	Birmingham, United Kingdom	3rd	4x400 m relay	3:08.14
	Universiade	Bangkok, Thailand	7th	400 m	46.85
			1st	4x400 m relay	3:02.05
2008	World Indoor Championships	Valencia, Spain	4th	4x400 m relay	3:08.76
	Olympic Games	Beijing, China	7th	4x400 m relay	3:00.32
2009	Universiade	Belgrade, Serbia	2nd	4x400 m relay	3:05.69

In what city did Piotr's last 1st place finish occur?

[Pasupat & Liang, 2015]

WikiTableQuestions dataset

Statistics:

- 22000 question/answers
- 2100 tables
- 6.3 columns and 27.5 rows per table

[Pasupat & Liang, 2015]

WikiTableQuestions dataset

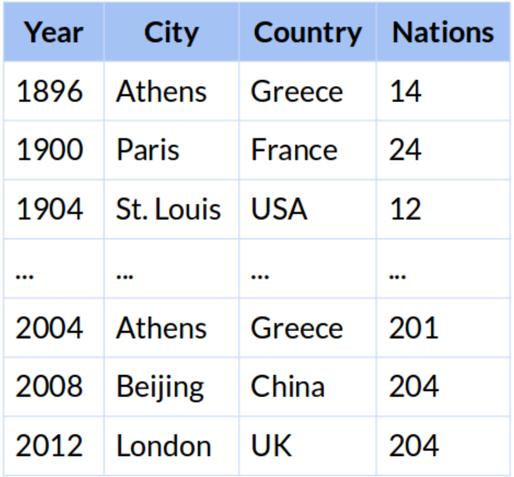
Statistics:

- 22000 question/answers
- 2100 tables
- 6.3 columns and 27.5 rows per table

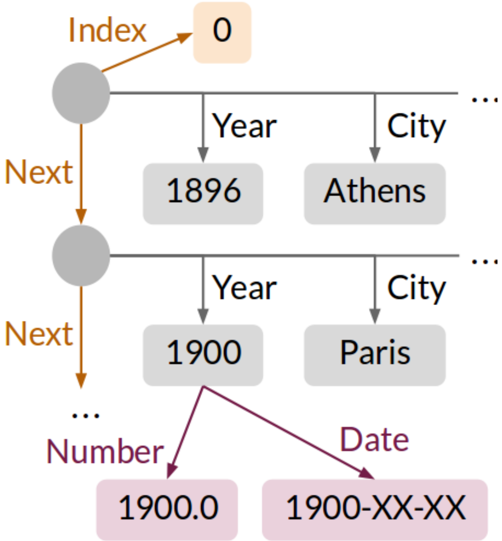
Challenges:

- High logical complexity (conjunction, disjunction, superlatives, comparatives, aggregation, arithmetic)
- Tables are unnormalized
- Train and test tables are distinct; need to generalize!

Model framework



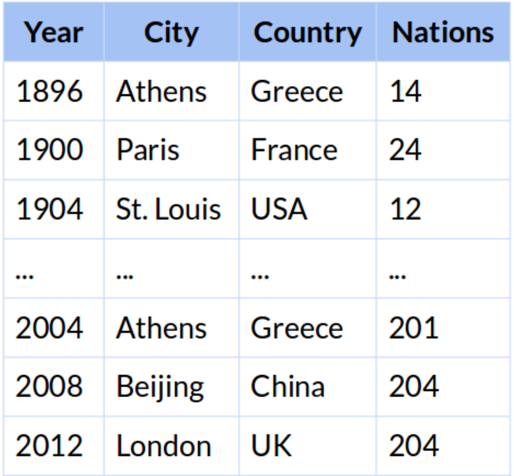
Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204



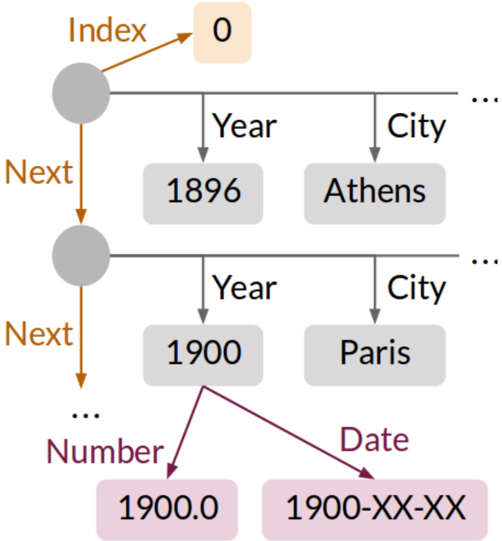
Greece held its last Summer Olympics in which year?

2004

Model framework



Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204



Greece held its last Summer Olympics in which year?

$$\downarrow$$

$$\mathbf{R}[\text{Date}].\mathbf{R}[\text{Year}].\text{argmax}(\text{Country.Greece}, \text{Index})$$

$$\downarrow$$

2004

Model details

Generate programs recursively of increasing size:

Greece

City

Country

Nations

Year

Model details

Generate programs recursively of increasing size:

Greece Country.Greece

City

Country

Nations

Year

Model details

Generate programs recursively of increasing size:

Greece	Country.Greece
City	$\mathbf{R}$[City].Country.Greece
Country	$\mathbf{R}$[Nations].Country.Greece
Nations	$\mathbf{R}$[Year].Country.Greece
Year	

Model details

Generate programs recursively of increasing size:

Greece

City

Country

Nations

Year

Country.Greece

$\mathbf{R}$[City].Country.Greece

$\mathbf{R}$[Nations].Country.Greece

$\mathbf{R}$[Year].Country.Greece

argmax(Country.Greece, Index)

argmax(Country.Greece, Nations)

argmax(Country.Greece, Year)

...

$\mathbf{R}$[Date].$\mathbf{R}$[Year].argmax(Country.Greece, Index)

Model details

Generate programs recursively of increasing size:

Greece
City
Country
Nations
Year

Country.Greece
$\mathbf{R}$[City].Country.Greece
$\mathbf{R}$[Nations].Country.Greece
$\mathbf{R}$[Year].Country.Greece

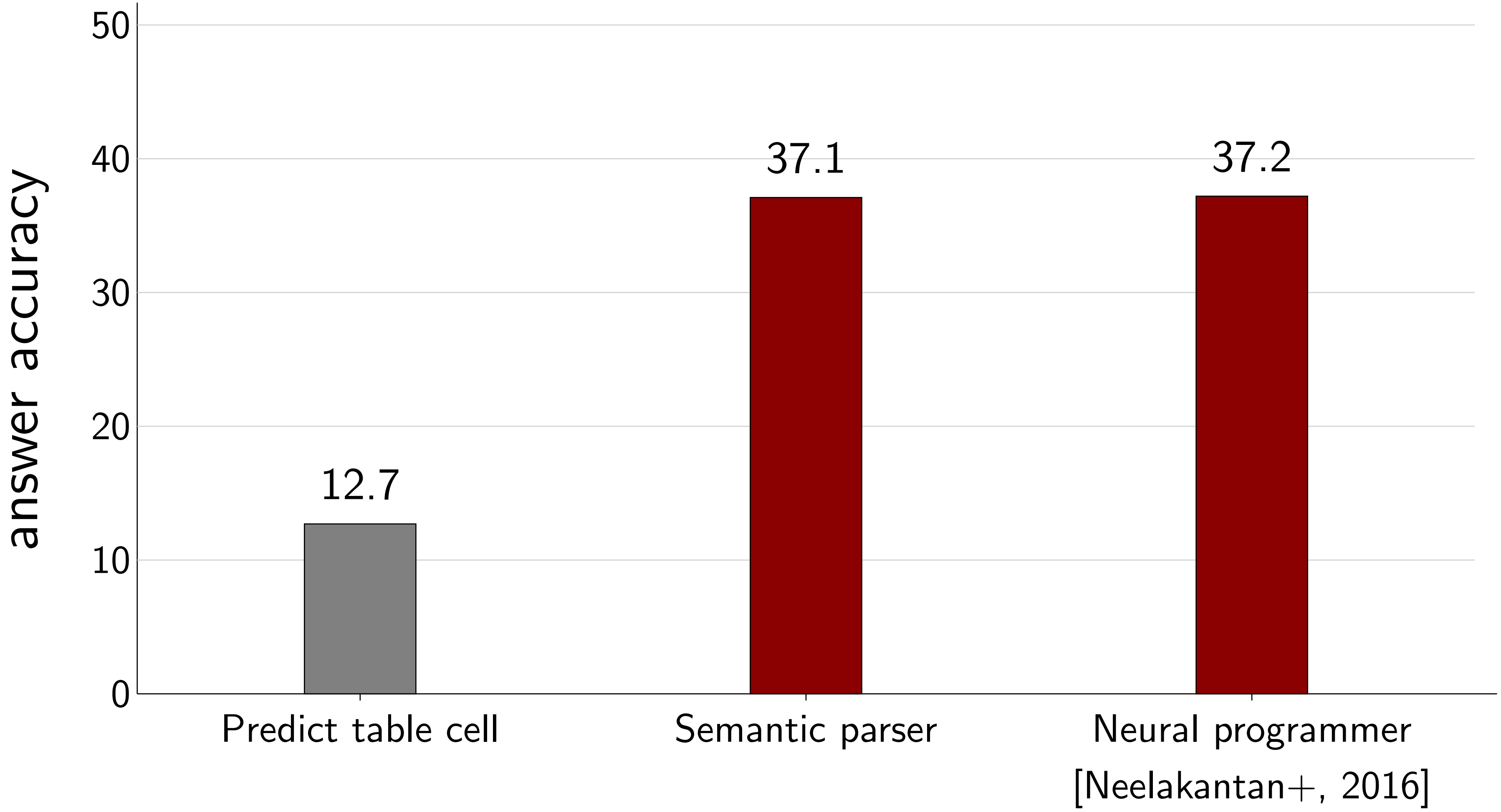
argmax(Country.Greece, Index)
argmax(Country.Greece, Nations)
argmax(Country.Greece, Year)
...

$\mathbf{R}$[Date].$\mathbf{R}$[Year].argmax(Country.Greece, Index)

Training:

$$\max_{\theta} \log \sum_{\text{Exec}(\text{program})=\text{answer}} p_{\theta}(\text{program} \mid \text{question})$$

Results on WikiTableQuestions



Error analysis

Unhandled operations:

- *Was there more gold medals won than silver?*
- *Which movies were number 1 for at least two* **consecutive** *weeks?*
- *How many titles had the* **same** *author listed as the illustrator?*

Error analysis

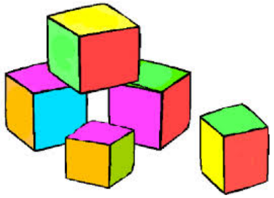
Unhandled operations:

- *Was there more gold medals won than silver?*
- *Which movies were number 1 for at least two* **consecutive** *weeks?*
- *How many titles had the* **same** *author listed as the illustrator?*

Table normalization:

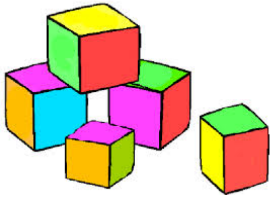
- *In what city did Piotr's last 1st place finish occur?* ...[Bangkok, Thailand]...
- *How long does the show defcon 3 last?* ...[2pm-3pm]...

Modularity (model)

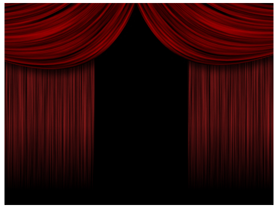


Natural language specifications

Moduarity (search)



Final remarks



Searching...

Greece held its last Summer Olympics in which year?

?

2004

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

Searching...

Greece held its last Summer Olympics in which year?

$\mathbf{R}[\text{Index}].\text{Country}.\text{Greece}$

2004

Year	**City**	**Country**	**Nations**
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

Searching...

Greece held its last Summer Olympics in which year?

$\mathbf{R}[\text{Nations}].\text{Country}.\text{Greece}$

2004

Year	**City**	**Country**	**Nations**
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

Searching...

Greece held its last Summer Olympics in which year?

argmax(Country.Greece, Nations)

2004

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

Searching...

Greece held its last Summer Olympics in which year?

$\text{argmax}(\text{Country.Greece}, \text{Index})$

2004

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

Searching...

Greece held its last Summer Olympics in which year?

... (thousands of logical forms later) ...

2004

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

Searching...

Greece held its last Summer Olympics in which year?

$\mathbf{R}[\text{Date}].\mathbf{R}[\text{Year}].\text{argmax}(\text{Country.Greece}, \text{Index})$

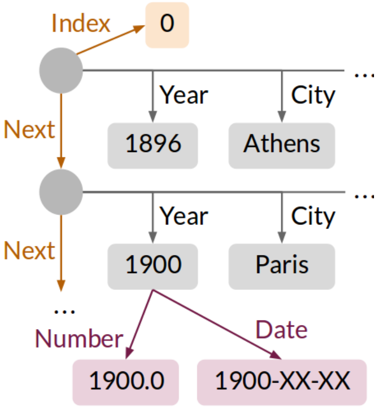
2004

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

Oracle accuracy

How many times did Greece hold the Summer Olympics?

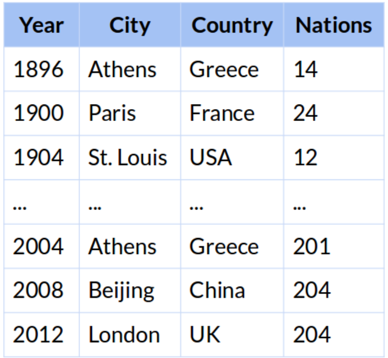
Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204



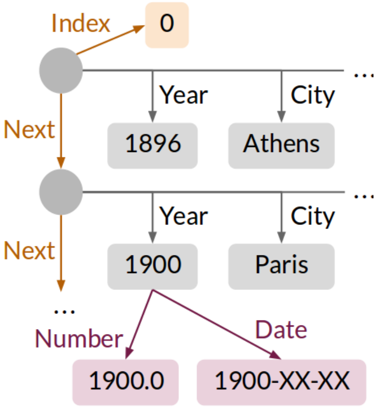
2

Oracle accuracy

How many times did Greece hold the Summer Olympics?



Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

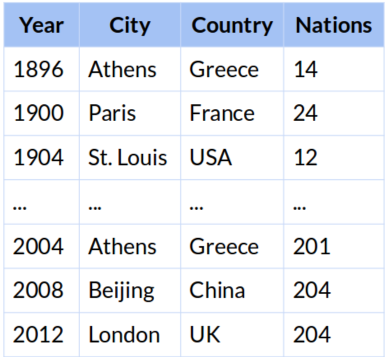


2

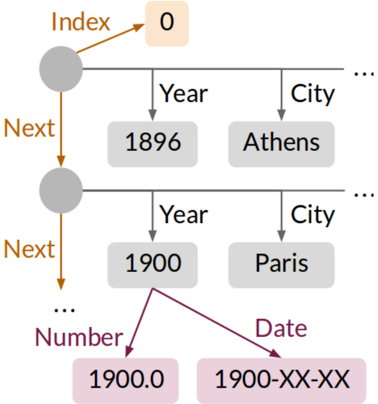
How often can the system even generate a set of 200 candidate programs containing the right answer?

Oracle accuracy

How many times did Greece hold the Summer Olympics?



Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204



2

How often can the system even generate a set of 200 candidate programs containing the right answer?

76.6%

Right for the wrong reasons

How many times did Greece hold the Summer Olympics?

count(Country.Greece)

2

Right for the wrong reasons

How many times did Greece hold the Summer Olympics?

count(Country.Greece) − count(Country.Norway)

2

Right for the wrong reasons

How many times did Greece hold the Summer Olympics?

$$\mathbf{R}[\text{Index}].\mathbf{R}[\text{Next}].\mathbf{R}[\text{Next}].\text{argmin}(\text{Country.Greece}, \text{Index})$$

2

Right for the wrong reasons

How many times did Greece hold the Summer Olympics?

$\mathbf{R}[\text{Index}].\mathbf{R}[\text{Next}].\mathbf{R}[\text{Next}].\text{argmin}(\text{Country.Greece}, \text{Index})$

2

System generates list of 200 candidate programs

Any gets correct answer: 76.6%

Any gets correct program: 53.5%

Right for the wrong reasons

How many times did Greece hold the Summer Olympics?

$\mathbf{R}[\text{Index}].\mathbf{R}[\text{Next}].\mathbf{R}[\text{Next}].\text{argmin}(\text{Country.Greece}, \text{Index})$

2

System generates list of 200 candidate programs

Any gets correct answer: 76.6%

Any gets correct program: 53.5%

Recovering program is unsupervised problem

Challenge

How many times did Greece hold the Summer Olympics?

2

Can we efficiently generate **all** programs (up to some size) that produce the correct answer?

[Pasupat & Liang, 2016]

Intuition: dynamic programming

$$\text{PopulationOf}.\text{CapitalOf}.\text{Colorado}$$

$$\text{PopulationOf}.\text{argmax}(\text{Type}.\text{City} \sqcap \text{ContainedBy}.\text{Colorado}, \text{Population})$$

↓

$$\text{PopulationOf}.\text{Denver}$$

[Pasupat & Liang, 2016]

Dynamic programming on denotations

Step 1: compute all reachable denotations

Colorado   Denver   United States   ...   649,495

[Pasupat & Liang, 2016]

Dynamic programming on denotations

Step 1: compute all reachable denotations

Colorado Denver United States ... 649,495

Step 2: discard denotations that don't reach the correct answer

(in practice: 99% reduction)

[Pasupat & Liang, 2016]

Dynamic programming on denotations

Step 1: compute all reachable denotations

Colorado Denver United States ... 649,495

Step 2: discard denotations that don't reach the correct answer

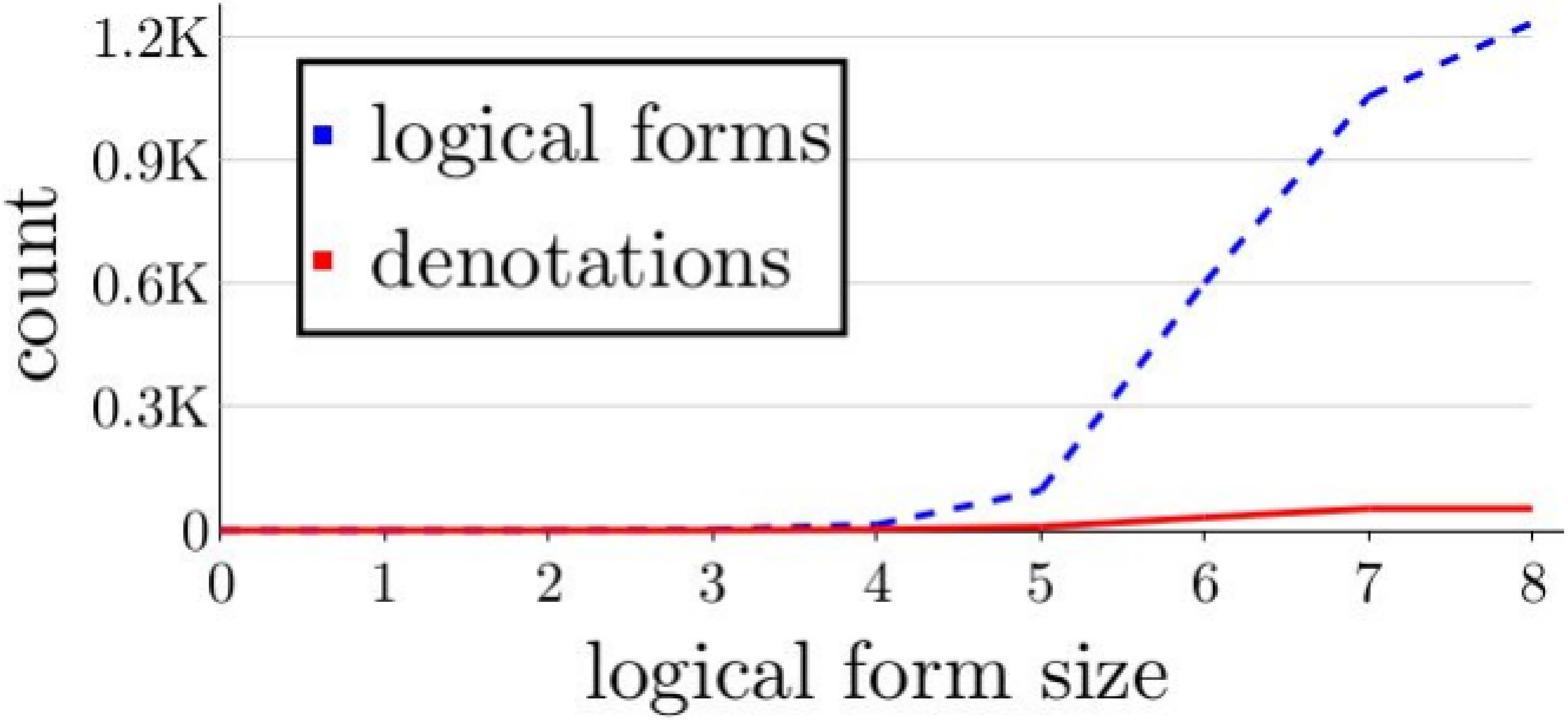
(in practice: 99% reduction)

Step 3: enumerate all programs on remaining denotations

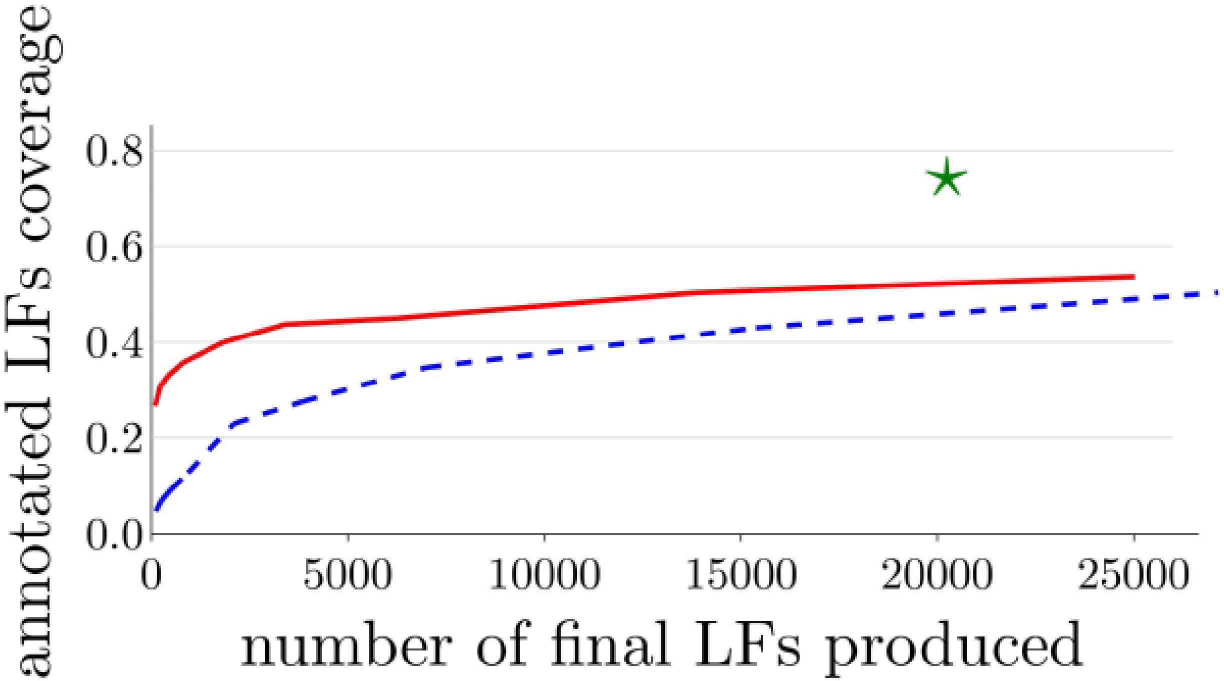
PopulationOf.CapitalOf.Colorado

PopulationOf.argmax(Type.City ⊓ ContainedBy.Colorado, Population)

Results: number of items explored



Results: oracle accuracy

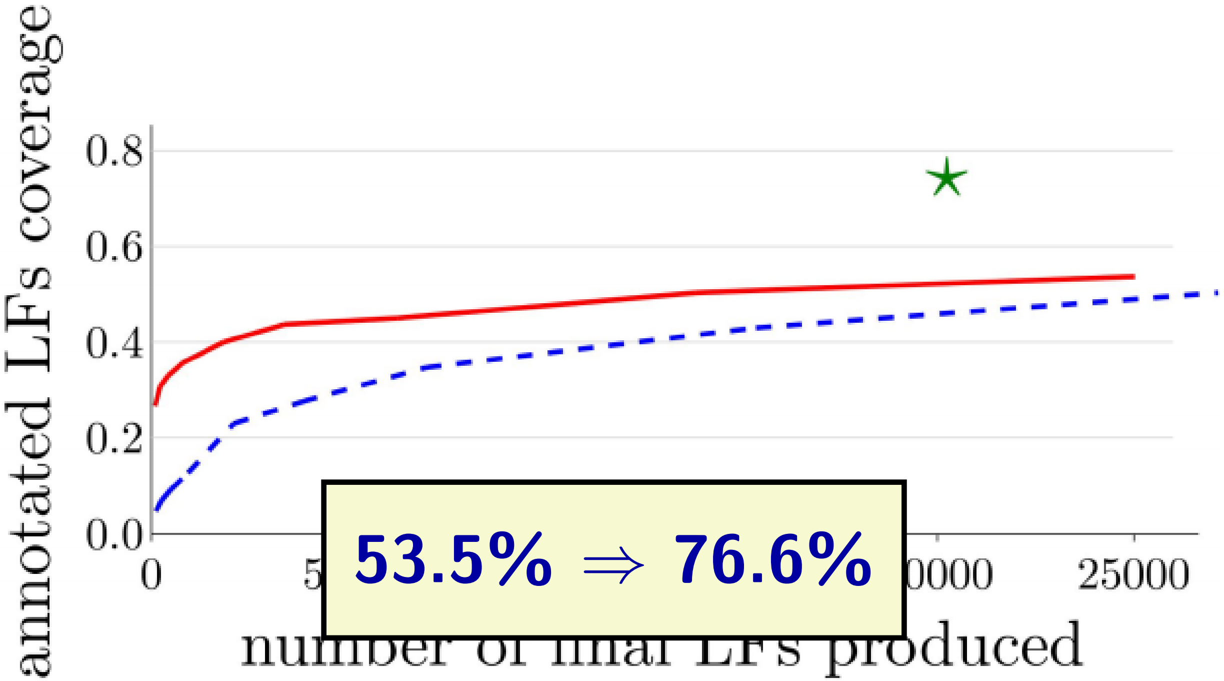


★ Dynamic programming on denotations

🟦 Uninitialized beam search

🟥 Initialized beam search

Results: oracle accuracy



53.5% ⇒ 76.6%

★ Dynamic programming on denotations

🟦 Uninitialized beam search

🟥 Initialized beam search

Learning macros

the same number of #0

    !=.cell1 AND R[col2].col0.R[col0].col2.cell1

how many more #0

    R[Number].R[col0].col2.cell1 - R[Number].R[col0].col2.cell3

at least NUM #0

    NUM ¿= @num #col:0

when was the last

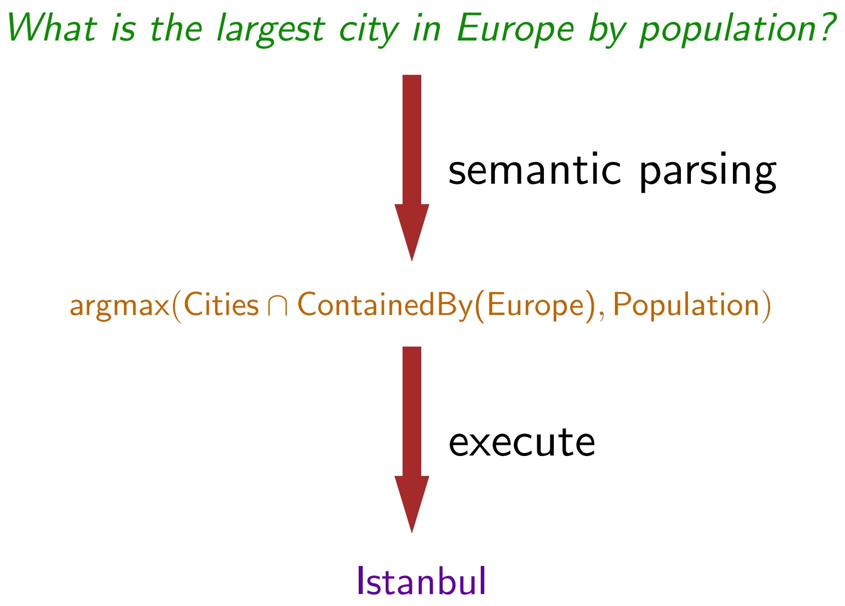
    max(R[Date].___)

after #0

    R[col1].R[Next].col1.cell0

Ongoing work: use these in learning the semantic parser

Summary so far



What is the largest city in Europe by population?

semantic parsing

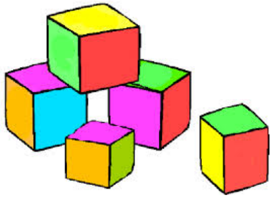
$\text{argmax}(\text{Cities} \cap \text{ContainedBy}(\text{Europe}), \text{Population})$

execute

Istanbul

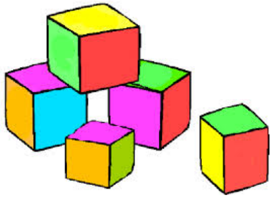
- Semantic parsing converts language to programs
- When we have learned the language, search is easy!
- Until then, search is hard, use dynamic programming

Modularity (model)

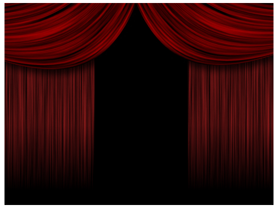


Natural language specifications

Moduarity (search)



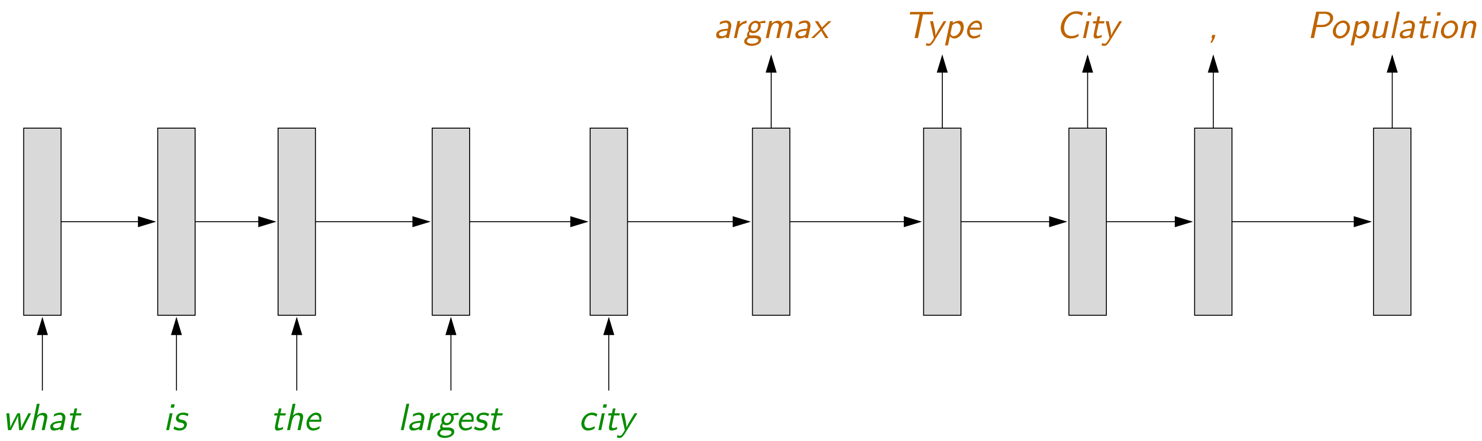
Final remarks



Where's the neural stuff?

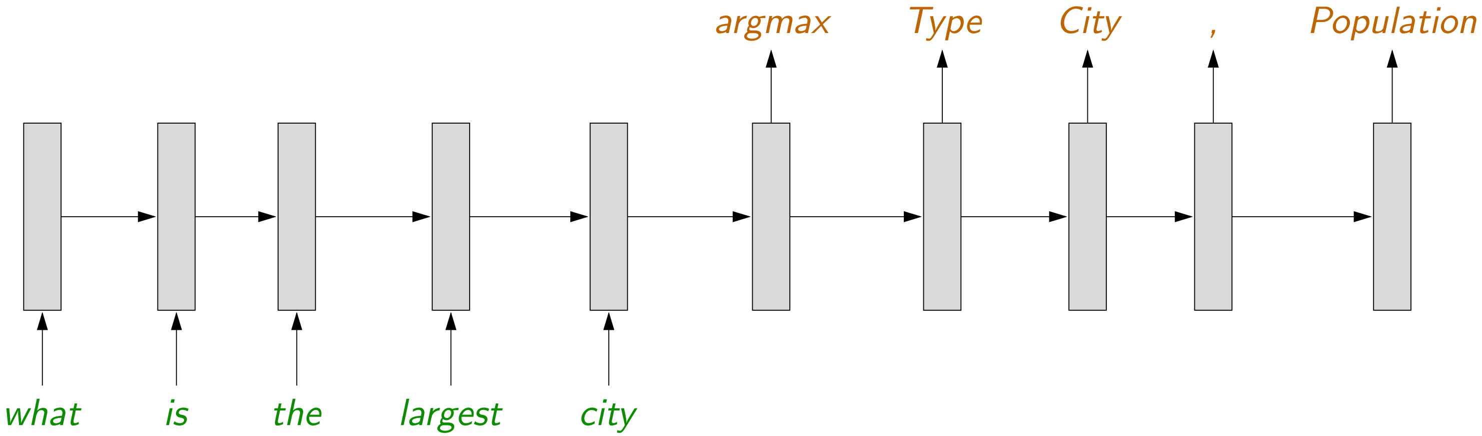
[Robin Jia, ACL 2016]

Neural semantic parsing



[Robin Jia, ACL 2016]

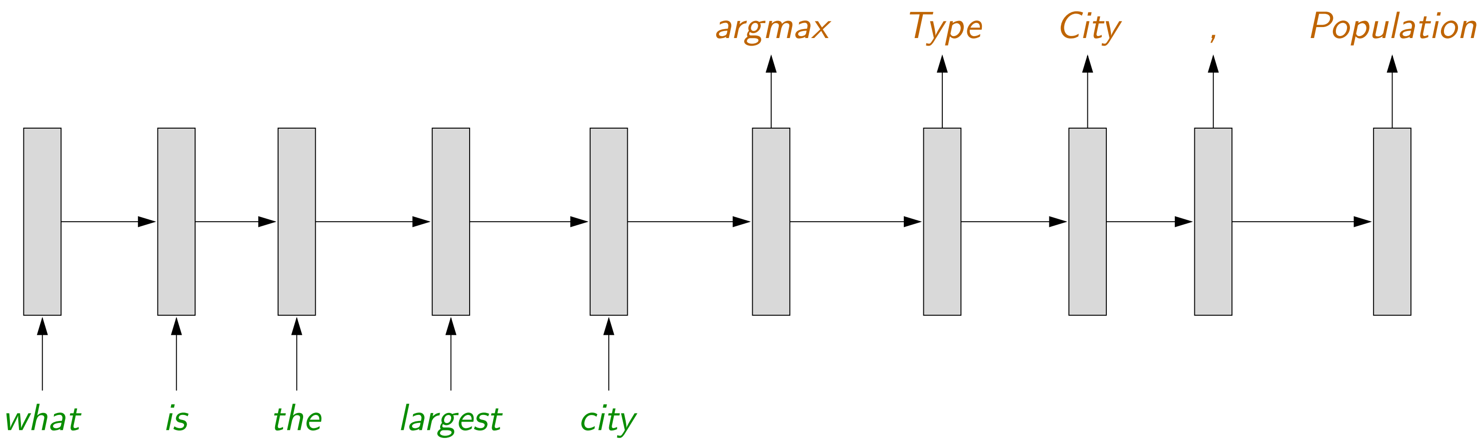
Neural semantic parsing



- Learn semantic composition without predefined grammar

[Robin Jia, ACL 2016]

Neural semantic parsing

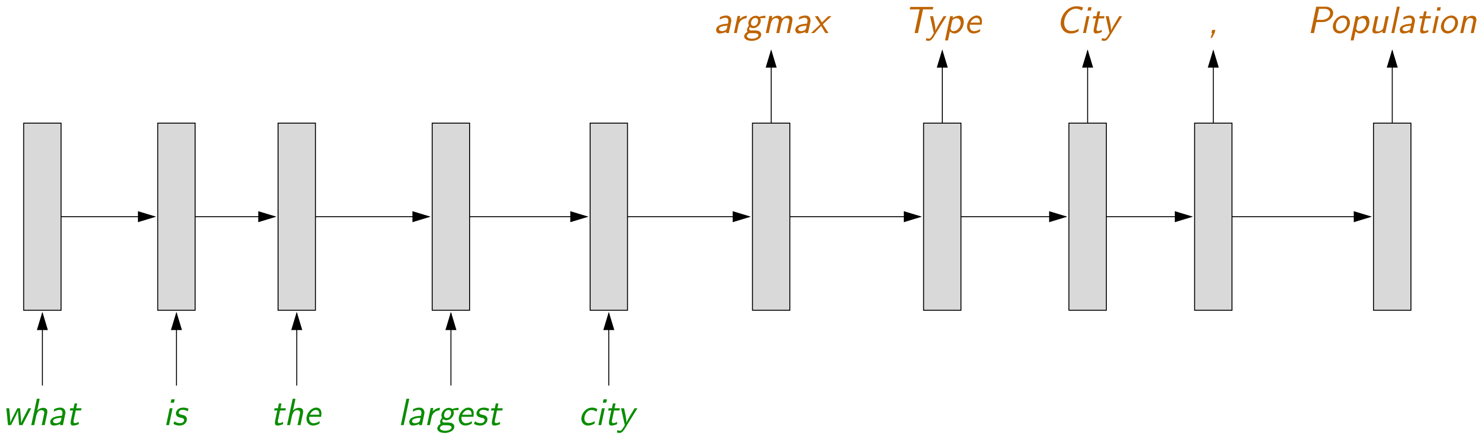


- Learn semantic composition without predefined grammar
- Encode compositionality through **data recombination**

what's the capital of Germany?	CapitalOf(Germany)
what countries border France?	Borders(France)

[Robin Jia, ACL 2016]

Neural semantic parsing



- Learn semantic composition without predefined grammar
- Encode compositionality through **data recombination**

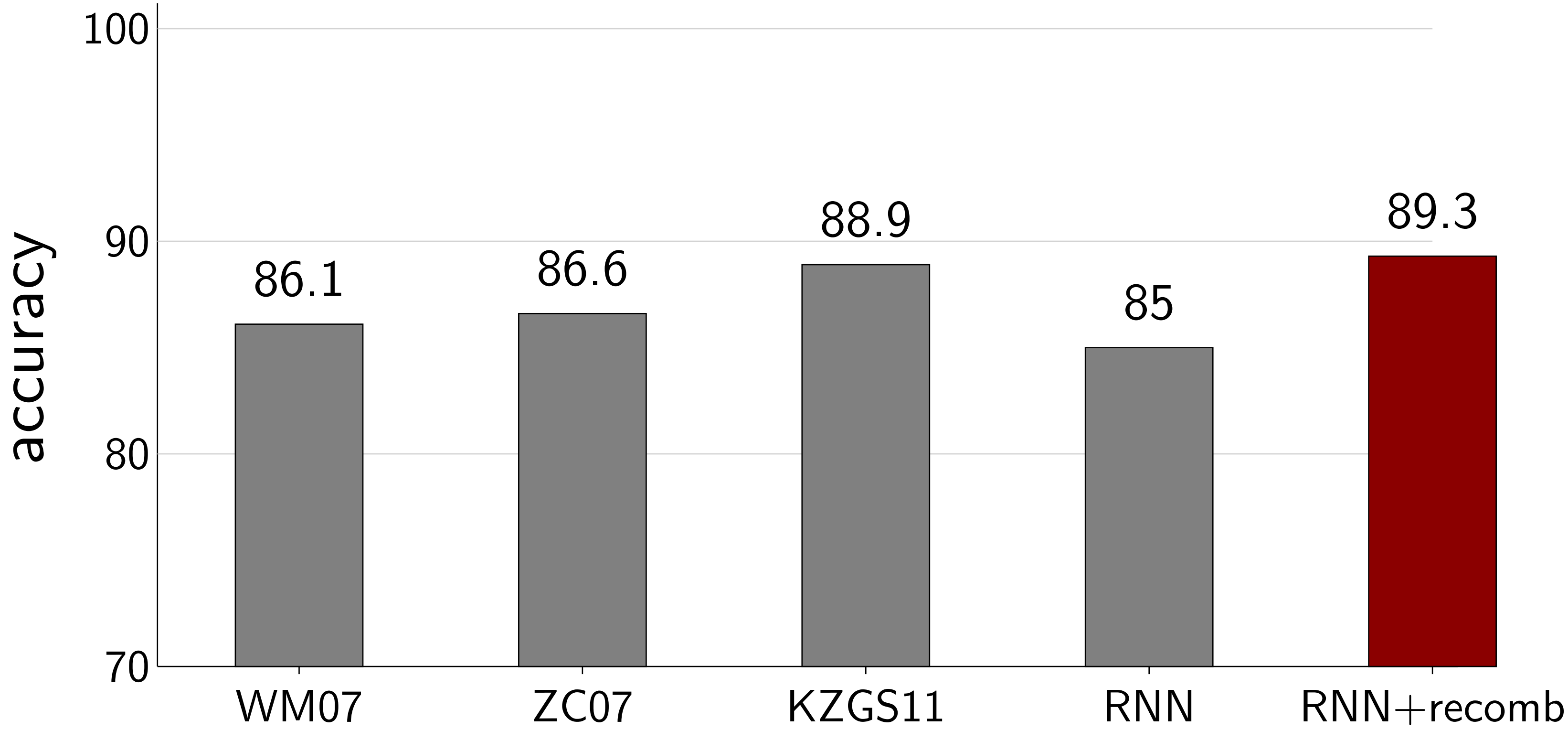
what's the capital of Germany?	CapitalOf(Germany)
what countries border France?	Borders(France)
what's the capital of France?	CapitalOf(France)
what countries border Germany?	Borders(Germany)

[Robin Jia, ACL 2016]

Neural semantic parsing

Dataset: US Geography dataset (Zelle & Mooney, 1996)

What is the highest point in Florida?



state-of-art, simpler

Point 1/2: factorization

question answering

Point 1/2: factorization

understanding

knowing

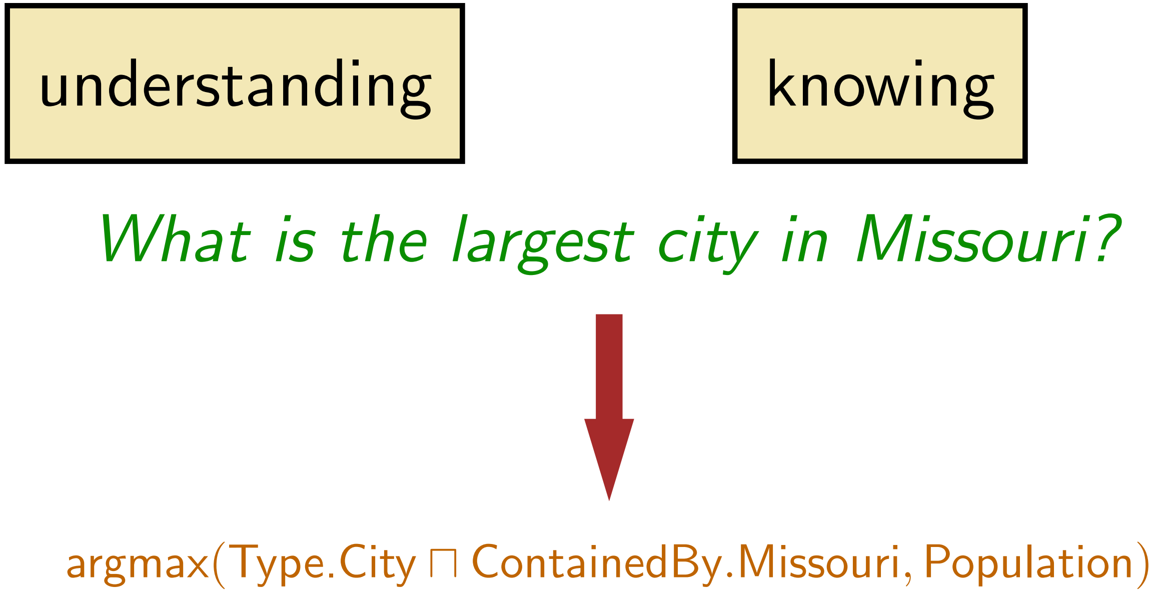
Point 1/2: factorization

understanding

knowing

What is the largest city in Missouri?

Point 1/2: factorization



understanding

knowing

What is the largest city in Missouri?

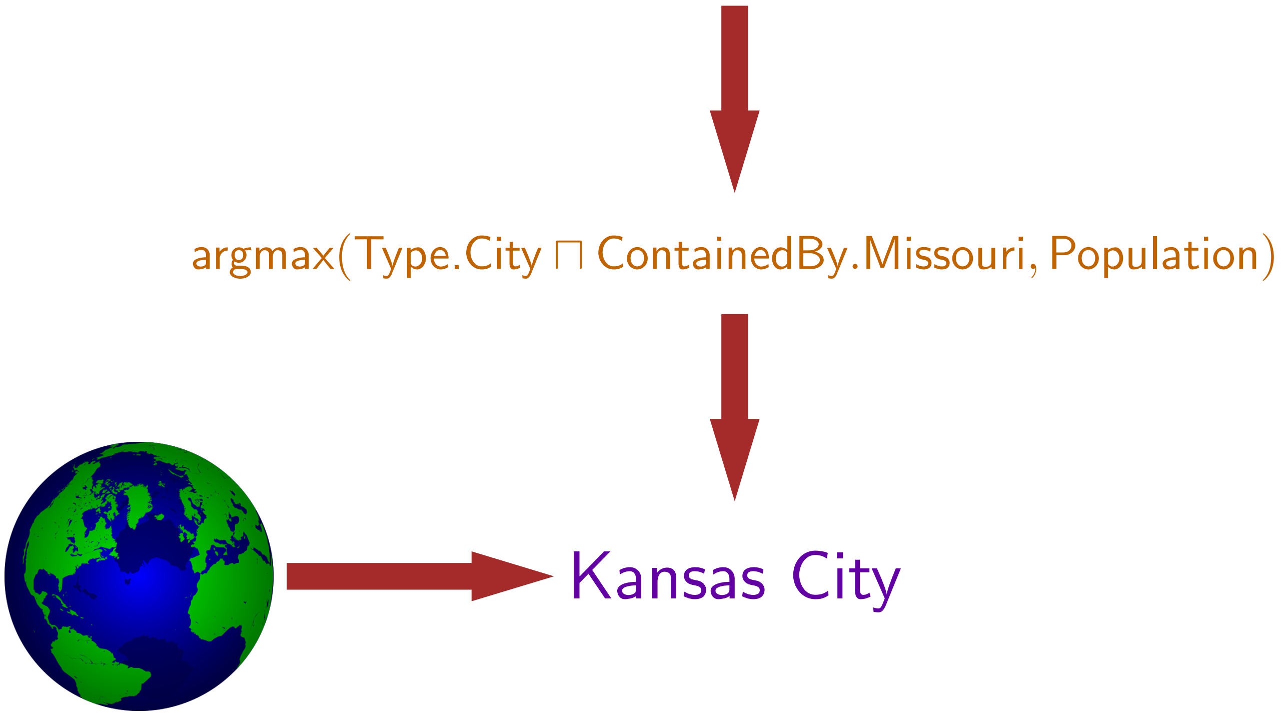
$\text{argmax}(\text{Type.City} \sqcap \text{ContainedBy.Missouri}, \text{Population})$

Point 1/2: factorization

understanding

knowing

What is the largest city in Missouri?



$\mathrm{argmax}(\mathrm{Type.City} \sqcap \mathrm{ContainedBy.Missouri}, \mathrm{Population})$

Kansas City

Point 1/2: factorization

understanding

knowing

What is the largest city in Missouri?

$\mathrm{argmax}(\mathrm{Type.City} \sqcap \mathrm{ContainedBy.Missouri}, \mathrm{Population})$

Kansas City

Generalize robustly to all worlds!

Point 2/2: discrete execution

What does differentiable execution buy you?

- Soft reasoning, perhaps (happens in language)
- Avoid combinatorial search

Point 2/2: discrete execution

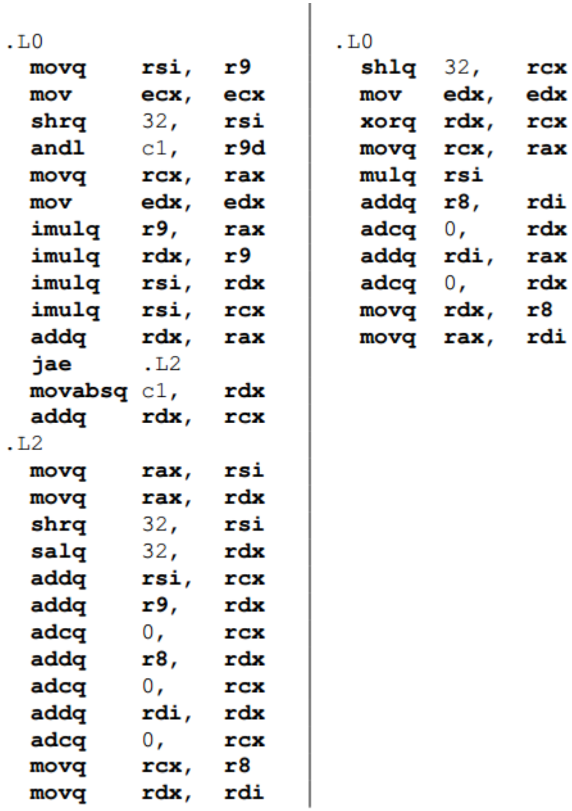
What does differentiable execution buy you?

- Soft reasoning, perhaps (happens in language)
- Avoid combinatorial search

But get nasty non-convex optimization problem instead!

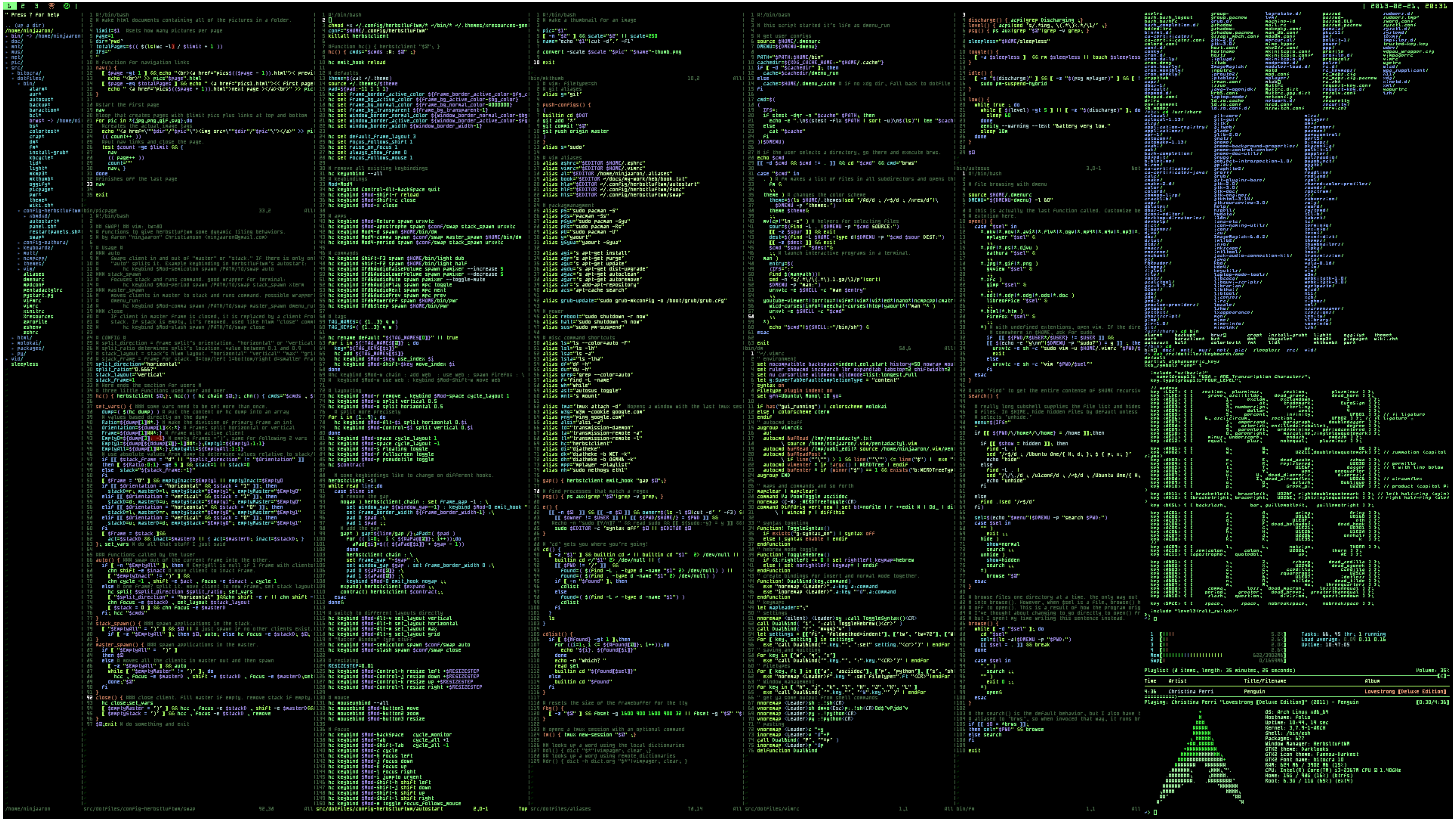
Point 2/2: discrete execution

- Real gains might be **overprovisioning** (many hidden units)
- Happens in discrete search too (many registers) [Schkufza+, 2013]



```
.L0
  movq     rsi,  r9
  mov      ecx,  ecx
  shrq     32,   rsi
  andl     c1,   r9d
  movq     rcx,  rax
  mov      edx,  edx
  imulq    r9,   rax
  imulq    rdx,  r9
  imulq    rsi,  rdx
  imulq    rsi,  rcx
  addq     rdx,  rax
  jae      .L2
  movabsq  c1,   rdx
  addq     rdx,  rcx
.L2
  movq     rax,  rsi
  movq     rax,  rdx
  shrq     32,   rsi
  salq     32,   rdx
  addq     rsi,  rcx
  addq     r9,   rdx
  adcq     0,    rcx
  addq     r8,   rdx
  adcq     0,    rcx
  addq     rdi,  rdx
  adcq     0,    rcx
  movq     rcx,  r8
  movq     rdx,  rdi
```

```
.L0
  shlq  32,   rcx
  mov   edx,  edx
  xorq  rdx,  rcx
  movq  rcx,  rax
  mulq  rsi
  addq  r8,   rdi
  adcq  0,    rdx
  addq  rdi,  rax
  adcq  0,    rdx
  movq  rdx,  r8
  movq  rax,  rdi
```



```
1  2  3                                                                                     | 2013-02-21, 20:31
```

```
" Press ? for help

.. (up a dir)
/home/ninjaaron/
▸ bin/ -> /home/ninjaaron
▸ doc/
▸ mnt/
▸ mus/
▸ net/
▸ pic/
▾ src/
  ▸ bitocra/
  ▾ dotfiles/
    ▾ bin/
        alarm*
        aur*
        autosus*
        backup*
        baraction*
        bcl*
        brws* -> /home/ni
        bs*
        colortest*
        crap*
        dm*
        fm*
        install-grub*
        kbcycle*
        lid*
        light*
        mkmp3*
        mkthumb*
        oggify*
        picpage*
        pwr*
        theme*
        wiki.sh*
    ▾ config-herbstluftwm
      ▸ xbm8x8/
        autostart*
        panel.sh*
        restartpanels.sh*
        swap*
    ▸ config-zathura/
    ▸ keyboardz/
    ▸ mutt/
    ▸ ncmpcpp/
    ▸ themes/
    ▸ vim/
      aliases
      dmenurc
      mpdconf
      pentadactylrc
      pystart.py
      vifmrc
      vimrc
      xinitrc
      Xresources
      zprofile
      zshenv
      zshrc
  ▸ html/
  ▸ molokai/
  ▸ packages/
  ▸ py/
▸ vid/
  sleepless
/home/ninjaaron
```

```
#!/bin/bash
# make html documents containing all of the pictures in a folder.

limit=$1  #sets how many pictures per page
page=1
dir=`pwd`
totalPages=$(( $(ls|wc -l) / $limit + 1 ))
IFS="
"
# function for navigation links
nav() {
  [ $page -gt 1 ] && echo "<br><a href="pics$(($page - 1)).html">< previ
    echo "<br>" >> pics"$page".html
  [ $page -eq $totalPages ] && echo " <a href="pics1.html"><< first page
    echo " <a href="pics$(($page + 1)).html">next page ></a><br>" >> pic
}

#start the first page
nav
#loop that creates pages with $limit pics plus links at top and bottom
for pic in *{jpg,png,gif,svg};do
  #creates the actual image tags
  echo "<a href=\""$dir"/"$pic"\"><img src=\""$dir"/"$pic"\"></a>" >> pi
  (( count++ ))
  #put nav links and close the page.
  test $count -ge $limit && {
    nav
    (( page++ ))
    count=""
    nav; }
done
#finishes off the last page
nav

exit
bin/picpage                                                   33,2          All
```

```
#!/bin/bash

## SWAP! ## vim: tw=80
# functions to give herbstluftwm some dynamic tiling behaviors.
# by Aaron "ninjaaron" Christianson <ninjaaron@gmail.com>

# Usage #
### auto
#    Swaps client in and out of "master" or "stack." If there is only on
#    "auto" splits it. Example keybinding in herbstluftwm's autostart:
#        hc keybind $Mod-semicolon spawn /PATH/TO/swap auto
### stack_spawn
#    focuses stack and runs command. good wrapper for terminal:
#        hc keybind $Mod-period spawn /PATH/TO/swap stack_spawn xterm
### master_spawn
#    moves clients in master to stack and runs command. possible wrapper
#    dmenu_run:
#        hc keybind $Mod-comma spawn /PATH/TO/swap master_spawn dmenu_ru
### close
#    If client in master frame is closed, it is replaced by a client fro
#    stack. If stack is empty, it's removed. used like hlwm "close" comm
#        hc keybind $Mod-slash spawn /PATH/TO/swap close

# CONFIG #
# split_direction = frame split's orientation. "horizontal" or "vertical
# split_ratio determines split's location. value between 0.1 and 0.9
# stack_layout = stack's hlwm layout. "horizontal" "vertical" "max" "gri
# stack_frame = frame for stack. 0=top/left 1=bottom/right d=smaller fra
split_direction="horizontal"
split_ratio="0.6667"
stack_layout="vertical"
stack_frame=1
# Here ends the section for users #
# three little functions used over and over.
hc() { herbstclient $@;}; hcc() { hc chain $@;}; chn() { cmds="$cmds , $

set_vars() { ### some vars need to be set more than once.
  dump=( $(hc dump) ) # put the content of hc dump into an array
  # values based directly on the dump
  Ratio=${dump[1]#*.} # make the division of primary frame an int
  Orientation=${dump[1]%:*} # frames split horizontal or vertical
  Frame=${dump[1]##*:} # frame with active client
  Empty0=${dump[3]: -1} # empty frames =')', same for following 2 vars
  Empty1=${dump[${#dump[@]}-1]##*:};Empty1=${Empty1:1:1}
  EmptyAll=${dump[1]#*:};EmptyAll=${EmptyAll:1:1}
  # use absolute values from dump to determine values relative to stack/
  if [[ $stack_frame = "d" || "$split_direction" != "$Orientation" ]]
  then [ ${Ratio:0:1} -ge 5 ] && stack=1 || stack=0
  else  stack="${stack_frame:-1}"
  fi
  [ $Frame = "0" ] && emptyInact=$Empty1 || emptyInact=$Empty0
  if [[ $Orientation = "horizontal" && $stack = "1" ]]; then
    stackD=r; masterD=l; emptyStack="$Empty1"; emptyMaster="$Empty0"
  elif [[ $Orientation = "vertical" && $stack = "1" ]]; then
    stackD=d; masterD=u; emptyStack="$Empty1"; emptyMaster="$Empty0"
  elif [[ $Orientation = "horizontal" && $stack = "0" ]]; then
    stackD=l; masterD=r; emptyStack="$Empty0"; emptyMaster="$Empty1"
  elif [[ $Orientation = "vertical" && $stack = "0" ]]; then
    stackD=u; masterD=d; emptyStack="$Empty0"; emptyMaster="$Empty1"
  fi
  [ $Frame = $stack ]&&
    act=$stackD && inact=$masterD || { act=$masterD; inact=$stackD; }
}; set_vars # do all that stuff I just said

### functions called by the luser
auto() { ### swap out of the current frame into the other.
  if [ -n "$EmptyAll" ]; then # EmptyAll is null if 1 frame with clients
    chn shift -e $inact # move client to inact frame.
    [ "$emptyInact" != ')' ] &&
    chn cycle -1 , shift -e $act , focus -e $inact , cycle 1
  else # one frame? split it. move client to new frame, set stack layout
    hc split $split_direction $split_ratio; set_vars
    [ "$split_direction" = "horizontal" ]&&chn shift -e r || chn shift -
    chn focus -e $stackD , set_layout $stack_layout
    [ $stack = 0 ] && chn focus -e $masterD
  fi; hcc "$cmds"
}
stack_spawn() { ### spawn applications in the stack.
  [ "$EmptyAll" = ')' ] && $@ || # just spawn if no other clients exist.
    if [ -z "$EmptyAll" ]; then $@; auto; else hc focus -e $stackD; $@;
}
master_spawn() { ### spawn applications in the master.
  if [ "$EmptyAll" =  ')' ]
  then $@
  else # moves all the clients in master out and then spawn
    [ -z "$EmptyAll" ] && auto
    while [ "$emptyMaster" != ')' ]; do
      hcc , focus -e $masterD , shift -e $stackD , focus -e $masterD;set
    done;"$@"
  fi
}
close() { ### close client. fill master if empty. remove stack if empty.
  hc close;set_vars
  [ $emptyMaster = ')' ] && hcc , focus -e $stackD , shift -e $masterD&&
  [ $emptyStack = ')' ] && hcc , focus -e $stackD , remove
}
$@;exit # do something and exit
src/dotfiles/config-herbstluftwm/swap                          92,38         All
```

```
#!/bin/bash
▯
chmod +x ~/.config/herbstluftwm/* ~/bin/* ~/.themes/xresources-gen
conf="$HOME/.config/herbstluftwm"
killall herbstclient

#function hc() { herbstclient "$@"; }
hc() { cmds="$cmds :#: $@" ;}

hc emit_hook reload

# defaults
theme=$(cat ~/.theme)
source ~/.themes/$theme
pad=${pad:-11 1 1 1}
hc set frame_border_active_color ${frame_border_active_color-$fg_c
hc set frame_bg_active_color ${frame_bg_active_color-$bg_color}
hc set frame_bg_normal_color ${frame_bg_normal_color-#000000}
hc set frame_bg_transparent ${frame_bg_transparent-1}
hc set window_border_normal_color ${window_border_normal_color-$bg
hc set window_border_active_color ${window_border_active_color-$fg
hc set window_border_width ${window_border_width-1}

hc set default_frame_layout 3
hc set focus_follows_shift 1
hc set raise_on_focus 1
hc set always_show_frame 0
hc set focus_follows_mouse 1

# remove all existing keybindings
hc keyunbind --all
# keybindings
Mod=Mod4
hc keybind Control-Alt-BackSpace quit
hc keybind $Mod-Shift-r reload
hc keybind $Mod-Shift-c close
hc keybind $Mod-x close

# apps
hc keybind $Mod-Return spawn urxvtc
hc keybind $Mod-apostrophe spawn $conf/swap stack_spawn urxvtc
hc keybind Mod4-d spawn $HOME/bin/dm
hc keybind Mod4-comma spawn $conf/swap master_spawn $HOME/bin/dm
hc keybind Mod4-period spawn $conf/swap stack_spawn urxvtc

# commands
hc keybind Shift-F3 spawn $HOME/bin/light dub
hc keybind Shift-F2 spawn $HOME/bin/light half
hc keybind XF86AudioRaiseVolume spawn pamixer --increase 5
hc keybind XF86AudioLowerVolume spawn pamixer --decrease 5
hc keybind XF86AudioMute spawn pamixer --toggle-mute
hc keybind XF86AudioPlay spawn mpc toggle
hc keybind XF86AudioNext spawn mpc next
hc keybind XF86AudioPrev spawn mpc prev
hc keybind XF86PowerOff spawn $HOME/bin/pwr
hc keybind XF86Sleep spawn $HOME/bin/pwr

# tags
TAG_NAMES=( {1..3} q w)
TAG_KEYS=( {1..3} q w )

hc rename default "${TAG_NAMES[0]}" || true
for i in ${!TAG_NAMES[@]} ; do
  key="${TAG_KEYS[$i]}"
  hc add ${TAG_NAMES[$i]}
  hc keybind $Mod-$key use_index $i
  hc keybind $Mod-Shift-$key move_index $i
done
#hc keybind $Mod-w chain : add web  : use web : spawn firefox : \
#  keybind $Mod-w use web : keybind $Mod-Shift-w move web

# layouting
hc keybind $Mod-r remove , keybind $Mod-space cycle_layout 1
hc keybind $Mod-u split vertical 0.5
hc keybind $Mod-o split horizontal 0.5
  # split more precisely
for i in {1..9}; do
  hc keybind $Mod-Alt-$i split horizontal 0.$i
  hc keybind $Mod-Control-$i split vertical 0.$i
done
hc keybind $Mod-space cycle_layout 1
hc keybind $Mod-space cycle_layout -1
hc keybind $Mod-s floating toggle
hc keybind $Mod-f fullscreen toggle
hc keybind $Mod-p pseudotile toggle
hc $contract

  # some keybindings like to change on different hooks.
herbstclient -i|
while read line;do
  case $line in
    # remove the gap
    nogap ) herbstclient chain : set frame_gap -1 : \
      set window_gap ${window_gap--1} : keybind $Mod-0 emit_hook "
      set frame_border_width ${frame_border_width-1} :\
      pad 0 $pad :\
      pad 1 $pad ;;
    # add the gap
    gap* ) gap=${line/gap /};aPad=( $pad )
      for (( i=0;  i < ${#aPad[@]}; i++));do
        aPad[$i]=$(( ${aPad[$i]} + $gap - 1))
      done
      herbstclient chain : \
      set frame_gap "-$gap" :\
      set window_gap $gap : set frame_border_width 0 :\
      pad 0 ${aPad[@]} :\
      pad 1 ${aPad[@]} :\
      keybind $Mod-0 emit_hook nogap ;;
    expand) herbstclient $expand ;;
    contract) herbstclient $contract;;
  esac
done&

# switch to different layouts directly
hc keybind $Mod-Alt-v set_layout vertical
hc keybind $Mod-Alt-h set_layout horizontal
hc keybind $Mod-Alt-m set_layout max
hc keybind $Mod-Alt-g set_layout grid
# "Master Window" type stuff
hc keybind $Mod-semicolon spawn $conf/swap auto
hc keybind $Mod-slash spawn $conf/swap close

# resizing
RESIZESTEP=0.01
hc keybind $Mod-Control-h resize left +$RESIZESTEP
hc keybind $Mod-Control-j resize down +$RESIZESTEP
hc keybind $Mod-Control-k resize up +$RESIZESTEP
hc keybind $Mod-Control-l resize right +$RESIZESTEP

# mouse
hc mouseunbind --all
hc mousebind $Mod-Button1 move
hc mousebind $Mod-Button2 zoom
hc mousebind $Mod-Button3 resize

# focus
hc keybind $Mod-BackSpace   cycle_monitor
hc keybind $Mod-Tab         cycle_all +1
hc keybind $Mod-Shift-Tab   cycle_all -1
hc keybind $Mod-c cycle
hc keybind $Mod-h focus left
hc keybind $Mod-j focus down
hc keybind $Mod-k focus up
hc keybind $Mod-l focus right
hc keybind $Mod-i jumpto urgent
hc keybind $Mod-Shift-h shift left
hc keybind $Mod-Shift-j shift down
hc keybind $Mod-Shift-k shift up
hc keybind $Mod-Shift-l shift right
hc keybind $Mod-m toggle focus_follows_mouse
src/dotfiles/config-herbstluftwm/autostart          2,0-1          Top
```

```
#!/bin/bash
# make a thumbnail for an image

pic="$1"
[ -n "$2" ] && scale="$2" || scale=250
name=`echo "$1"|cut -d'.' -f1`

convert -scale $scale "$pic" "$name"-thumb.png

exit
bin/mkthumb                                                10,2          All
```

```
# vim: filetype=sh
# git aliases
alias g='git'

push-configs() {
(
builtin cd $DOT
git add '*'
git commit "$@"
git push origin master
)
}
alias s='sudo'

# vim aliases
alias zshrc="$EDITOR $HOME/.zshrc"
alias vimrc="$EDITOR $HOME/.vimrc"
alias al="$EDITOR /home/ninjaaron/.aliases"
alias book="$EDITOR ~/docs/my-work/heb/book.txt"
alias hla="$EDITOR ~/.config/herbstluftwm/autostart"
alias hlf="$EDITOR ~/.config/herbstluftwm/func"
alias hls="$EDITOR ~/.config/herbstluftwm/swap"

# packagmanagment
alias pS="sudo pacman -S"
alias pSs="pacman -Ss"
alias pSyu="sudo pacman -Syu"
alias pRs="sudo pacman -Rs"
alias pU="sudo pacman -U"
alias y="yaourt"
alias ySyua="yaourt -Syua"

alias agi='s apt-get install'
alias agp='s apt-get purge'
alias agu='s apt-get update'
alias agdu='s apt-get dist-upgrade'
alias agac='s apt-get autoclean'
alias agar='s apt-get autoremove'
alias aar='s add-apt-repository'
alias acs='apt-cache search'

alias grub-update="sudo grub-mkconfig -o /boot/grub/grub.cfg"

# power
alias reboot="sudo shutdown -r now"
alias halt="sudo shutdown -h now"
alias sus="sudo pm-suspend"

# misc command shortcuts
alias ls="ls --color=auto -F"
alias lsl="ls -lh"
alias lsa="ls -a"
alias lsla="ls -lha"
alias df="df -h"
alias du="du -h"
alias grep='grep --color=auto'
alias f='find -L -name'
alias wh="while"
alias ast="autosus toggle"
alias mnt='s mount'

alias tma='tmux attach -d'  #opens a window with the last tmux ses
alias w3g='w3m -cookie google.com'
alias png="ping google.com"
alias alsi="alsi -a"
alias td="transmission-daemon"
alias ta="transmission-remote -a"
alias tl="transmission-remote -l"
alias hc="herbstclient"
alias di="diatheke"
alias dk="diatheke -b NET -k"
alias dh="diatheke -b OSMHB -k"
alias mpp="mplayer -playlist"
alias nh="sudo nethogs eth1"

gap() { herbstclient emit_hook "gap $@";}

# find processes that match a regex
psg() { ps aux|grep "$@"|grep -v grep; }

e() {
  [[ -n $@  ]] && [[ -e $@ ]] && owner=$(ls -l $@|cut -d' ' -f3) &
    [[ $owner != $USER ]] || [[ ${PWD/$HOME/} = $PWD ]] &&
   #echo -n "sudo [Y/n]? " && read sudo && [[ ${sudo:-y} = y ]] &&
    sudo $EDITOR -c 'syntax off' $@ || $EDITOR $@
}

# `cd` gets you where you're going!
cd() {
  [ -z "$1" ] && builtin cd ~ || builtin cd "$1"  2> /dev/null ||
    builtin cd ~/"$1" 2> /dev/null || {
    [[ $PWD != '/' ]]  &&
      found=( $(find -L . -type d -name "$1" 2> /dev/null) ) ||
      found=( $(find . -type d -name "$1" 2> /dev/null) )
    if [ -n "$found" ]; then
      cdlist
    else
      found=( $(find -L ~ -type d -name "$1") )
      cdlist
    fi
  }
  ls
}

cdlist() {
  if [ ${#found} -gt 1 ];then
    for ((i=1; i <= ${#found[@]}; i++));do
      echo "${i}. ${found[$i]}"
    done
    echo -n "which? "
    read sel
    builtin cd "${found[$sel]}"
  else
    builtin cd "$found"
  fi
}

# resets the size of the framebuffer for the tty
fbg() {
  [ -z "$@" ] && fbset -g 1600 900 1600 900 32 || fbset -g "$@" "$

# opens a tmux session with an optional command
tm() { tmux new-session "$@" ;}

## looks up a word using the local dictionaries
#dl() { dict "$*"|vimpager; clear ;}
## looks up a word using remote dictionaries
#dr() { dict -h dict.org "$*"|vimpager; clear; }
src/dotfiles/aliases                                       78,14         All
```

```
#!/bin/bash

# this script started it's life as dmenu_run

# get user configs
source $HOME/.dmenurc
DMENU=${DMENU-dmenu}

PATH="$PATH:$HOME/bin"
cachedir=${XDG_CACHE_HOME:-"$HOME/.cache"}
if [ -d "$cachedir" ]; then
  cache=$cachedir/dmenu_run
else
  cache=$HOME/.dmenu_cache # if no xdg dir, fall back to dotfile
fi

cmd=$(
(
  IFS=:
  if stest -dqr -n "$cache" $PATH; then
    echo -e ".\n$(stest -flx $PATH | sort -u)\n$(ls)"| tee "$cach
  else
    cat "$cache"
  fi
)|$DMENU)

# if the user selects a directory, go there and execute brws.
echo $cmd
[[ -d $cmd && $cmd != . ]] && cd "$cmd" && cmd="brws"

case "$cmd" in
  . ) # fm makes a list of files in all subdirectors and opens th
    fm &
    ;;
  theme ) # changes the color scheme
    theme=$(ls $HOME/.themes|sed '/#d/d ; /~$/d ; /xres/d'|\
      $DMENU -p 'themes:')
    theme $theme&
    ;;
  mv|cp|"ln -s" ) # helpers for selecting files
    sour=$(find -L . |$DMENU -p "$cmd SOURCE:")
    [[ -z $sour ]] && exit
    dest=$(find -L $HOME -type d|$DMENU -p "$cmd $sour DEST:")
    [[ -z $dest ]] && exit
    $cmd "$sour" "$dest"&
    ;; # launch interactive programs in a terminal.
  man )
    entry=$(
    (IFS=':'
    find $(manpath))|
    sed -n 's/.*\/\(.*\).1.gz/\1/p'|sort|
    $DMENU -p "man:")
    urxvtc -e $SHELL -c "man $entry"
    ;;
  youtube-viewer*|torrtux*|vifm*|vim*|vi*|ed*|nano*|ncmpcpp|cmatr
    wicd-curses|info*|weechat-curses|htop|yaourt*|"man "* )
    urxvt -e $SHELL -c "$cmd"
    ;;
  *)
    echo "$cmd"|${SHELL:-"/bin/sh"} &
esac

exit
bin/dm                                                     58,6          All
```

```
"~/.vimrc
" environment
set nocompatible backspace=indent,eol,start history=50 nowrap mou
set ruler showcmd incsearch lbr expandtab tabstop=2 shiftwidth=2
set nu cursorline wildmenu wildmode=list:longest,full
let g:SuperTabDefaultCompletionType = "context"
syntax on
filetype plugin indent on
set gfn=Ubuntu\ Mono\ 10 go=

if has("gui_running") | colorscheme molokai
else | colorscheme cterm
endif
" autocmd stuff
augroup vimrcEx
  au!
  autocmd BufRead /tmp/pentadacty*.txt
        \ source /home/ninjaaron/.vim/pentadactyl.vim
  autocmd BufRead /tmp/uzbl_edit* source /home/ninjaaron/.vim/pen
  autocmd BufReadPost *
        \ if line("'\"") > 1 && line("'\"") <= line("$") |  exe "
  autocmd vimenter * if !argc() | NERDTree | endif
  autocmd bufenter * if (winnr("$") == 1 && exists("b:NERDTreeTyp
augroup END

" maps and commands and so forth
mapclear | mapclear!
command Va VoomToggle asciidoc
noremap <C-N> :NERDTreeToggle<CR>
command DiffOrig vert new | set bt=nofile | r ++edit # | 0d_ | di
      \ | wincmd p | diffthis

" syntax toggling
function! ToggleSyntax()
  if exists("g:syntax_on") | syntax off
  else | syntax enable | endif
endfunction
" hebrew mode toggle
function! ToggleHebrew()
  if &l:rightleft == 0 | set rightleft keymap=hebrew
  else | set norightleft keymap= | endif
endfunction
" create bindings for insert and normal mode together.
function! DualBind(key,command)
  exe "noremap <Leader>".a:key a:command
  exe "inoremap <Leader>".a:key "^O".a:command
endfunction
" keymaps
let mapleader=";"
" settings
nnoremap <silent> <Leader>sy :call ToggleSyntax()<CR>
call DualBind( 'i', ':call ToggleHebrew()<cr>' )
call DualBind( 'r', 'mvgq}`v' )
let settings = [['fi', 'foldmethod=indent'], ['tw', 'tw=72'], ['W
for [ key, setting ] in settings
  exe "call DualBind('".key."', ':set" setting."<cr>')" | endfor
" saving and quitting
for key in ['w', 'q', 'x']
  exe "call DualBind('".key."', ':".key."<CR>')" | endfor
" filetypes
for [ key, ft ] in [['a', 'asciidoc'], ['p', 'python'], ['s', 'sh
  exe "noremap <Leader>f".key " :set filetype=".ft "<CR>"|endfor
" Window management
for key in [ "h", "j", "k", "l", "H", "J", "K", "L" ]
  exe "call DualBind( '".key."', '^W".key."' )" | endfor
" get ya some output from shell commands
nnoremap <Leader>sh :.!sh<CR>
vnoremap <Leader>sh dmvo<Esc>p:.!sh<CR>0d$`vPjdd`v
nnoremap <Leader>py :.!python<CR>
vnoremap <Leader>py :!python<CR>
" pasting
vnoremap <Leader>c "+y
inoremap <Leader>v ^O"+P
call DualBind( 'P', '"*p' )
inoremap <Leader>p ^Op
delfunction DualBind
src/dotfiles/vimrc                                         1,1           All
```

```

discharge() { acpi|grep Discharging ;}
level() { acpi|sed 's/.*ing, \(.*\)%.*/\1/' ;}
psg() { ps aux|grep "$@"|grep -v grep; }

sleepless="$HOME/sleepless"

toggle() {
  [ -a $sleepless ]  && rm $sleepless || touch $sleepless
}

idle() {
  [ -n "$(discharge)" ] && [ -z "$(psg mplayer)" ] && [ !
    sudo pm-suspend-hybrid
}

low() {
  while true ; do
    while [ $(level) -gt 5 ] || [ -z "$(discharge)" ]; do
      sleep 60
    done
    zenity --warning --text "Battery very low."
    sleep 10m
  done
}

$@
bin/autosus                                                3,0-1         Bot
```

```
#!/bin/bash

# file browsing with dmenu

source $HOME/.dmenurc
DMENU="${DMENU-dmenu} -l 60"

# this is actually the last function called. Customize be
# extntion here.
open() {
  case "$sel" in
    *.mkv|*.mov|*.avi*|*.flv*|*.ogv|*.mp4*|*.m4v|*.mp3|*.
      mplayer "$sel" &
      ;;
    *.pdf|*.ps|*.djvu )
      zathura "$sel" &
      ;;
    *.jpg|*.gif|*.png )
      gqview "$sel" &
      ;;
    *.xfc )
      gimp "$sel" &
      ;;
    *.odt|*.odp|*.odg|*.ods|*.doc )
      libreoffice "$sel" &
      ;;
    *.html|*.htm )
      firefox "$sel" &
      ;;
    *) # with undefined extentions, open vim. If the dire
       # somewhere in $HOME, ask for sudo.
      if  [[ ${PWD/*$USER*/$USER} != $USER ]] &&
      [[ $(echo -e "y\nn"|$DMENU -p "sudo?") = y ]] ; the
        urxvtc -e sh -c "sudo vim -u $HOME/.vimrc '$PWD/$
        exit
      else
        urxvtc -e sh -c "vim '$PWD/$sel'"
      fi
  esac
}

# use `find` to get the entire contense of $HOME recursiv
search() {

  # really long subshell supplies the file list and hides
  # files. In $HIME, hide hidden files by defautl unless
  # selects 'unhide.'
  menu=$(IFS="
"
  if [[ ${PWD/\/home*/\/home} = /home ]];then

    if [[ $show = hidden ]]; then
      find -L .|
      sed '/~$/d ; /Ubuntu One/{ H; d; }; $ { p; x; }'
      echo "hide"
    else
      find -L . |
      sed "/\/\./d ; /ulconf/d ; /~$/d ; /Ubuntu One/{ H;
      echo "unhide"
    fi

  else
    find .|sed '/~$/d'
  fi)

  sel=$(echo "$menu"|$DMENU -p "search $PWD:")
  case $sel in
    "" )
      exit ;;
    hide )
      show=normal
      search ;;
    unhide )
      show=hidden
      search ;;
    *)
      browse "$@"
  esac
}

# browse files one directory at a time. the only way out
# into browse(). However, when $sel is a file, browse() h
# off to open(). This is a result of how the program orig
# I've thought about changing to go directly to open() fr
# but I spent my time writing this sentence instead.
browse() {
  while [ -d "$sel" ]; do
    cd "$sel"
    sel=$(ls -a|$DMENU -p "$PWD:")
    [[ $sel = . ]] && break
  done

  case $sel in
    "." )
      search ;;
    "" )
      exit 0 ;;
    *)
      open&
  esac
}

# the search() is the default behavior, but I also have t
# aliased to `brws`, so when invoced that way, it runs br
if [[ $0 = *brws ]];
then sel="$PWD" && browse
else search
fi

exit
bin/fm                                                     1,1           All
```

[illegible]

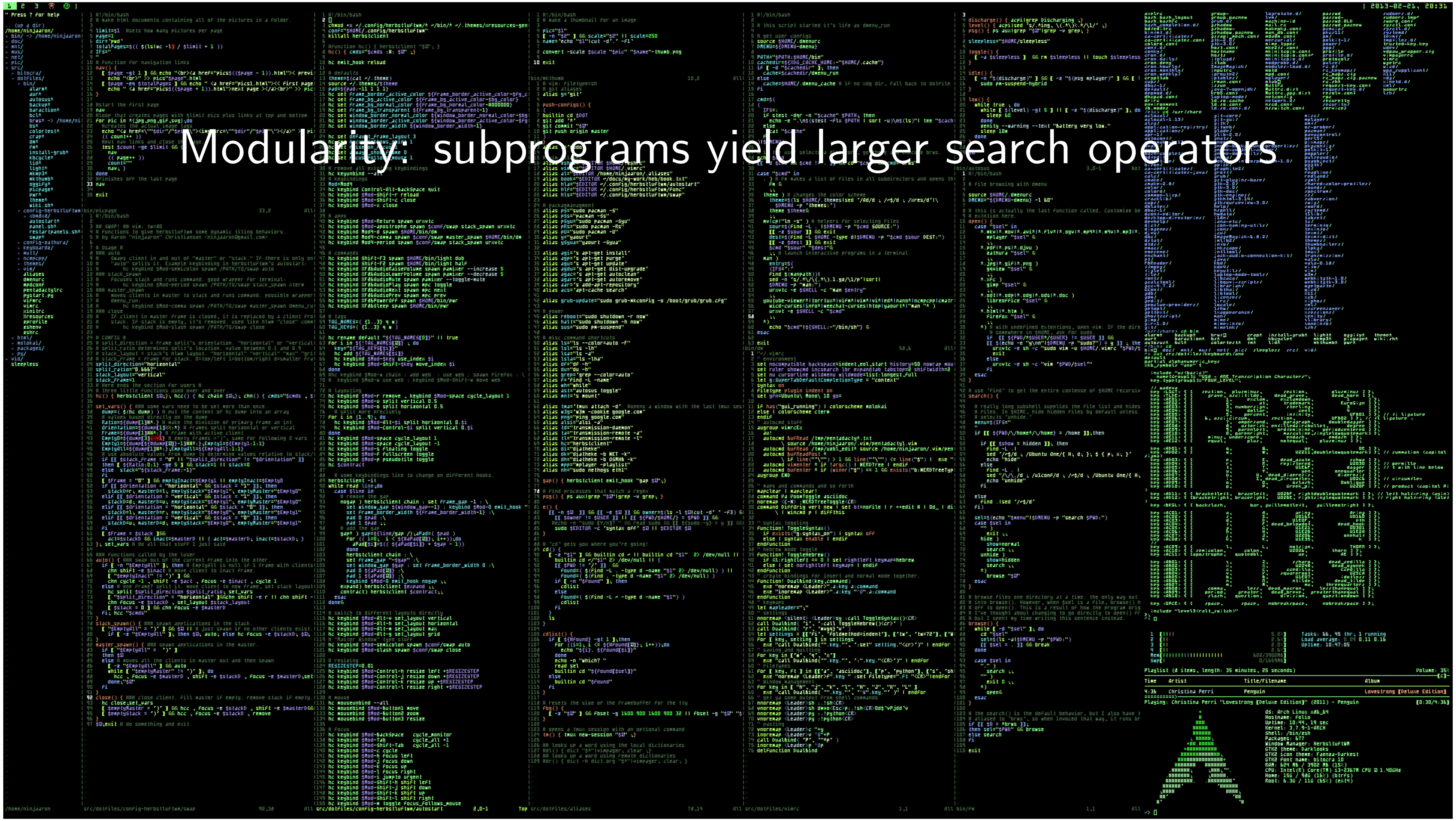
```
    key <AB02> { [          x,          X,              U2248,       dead_ogonek ] };
    key <AB03> { [          c,          C,           ccedilla,          Ccedilla ] };
    key <AB04> { [          v,          V,         squareroot,             U25CA ] };
    key <AB05> { [          b,          B,              U1E07,          idotless ] };
    key <AB06> { [          n,          N,             ntilde,        dead_tilde ] };
    key <AB07> { [          m,          M,                 mu,     threequarters ] };
    key <AB08> { [      comma,       less,        dead_macron,     lessthanequal ] };
    key <AB09> { [     period,    greater,         dead_breve,  greaterthanequal ] };
    key <AB10> { [      slash,   question,           division,      questiondown ] };

    key <SPCE> { [      space,      space,       nobreakspace,      nobreakspace ] };

    include "level3(ralt_switch)"
};
~> ▯
```

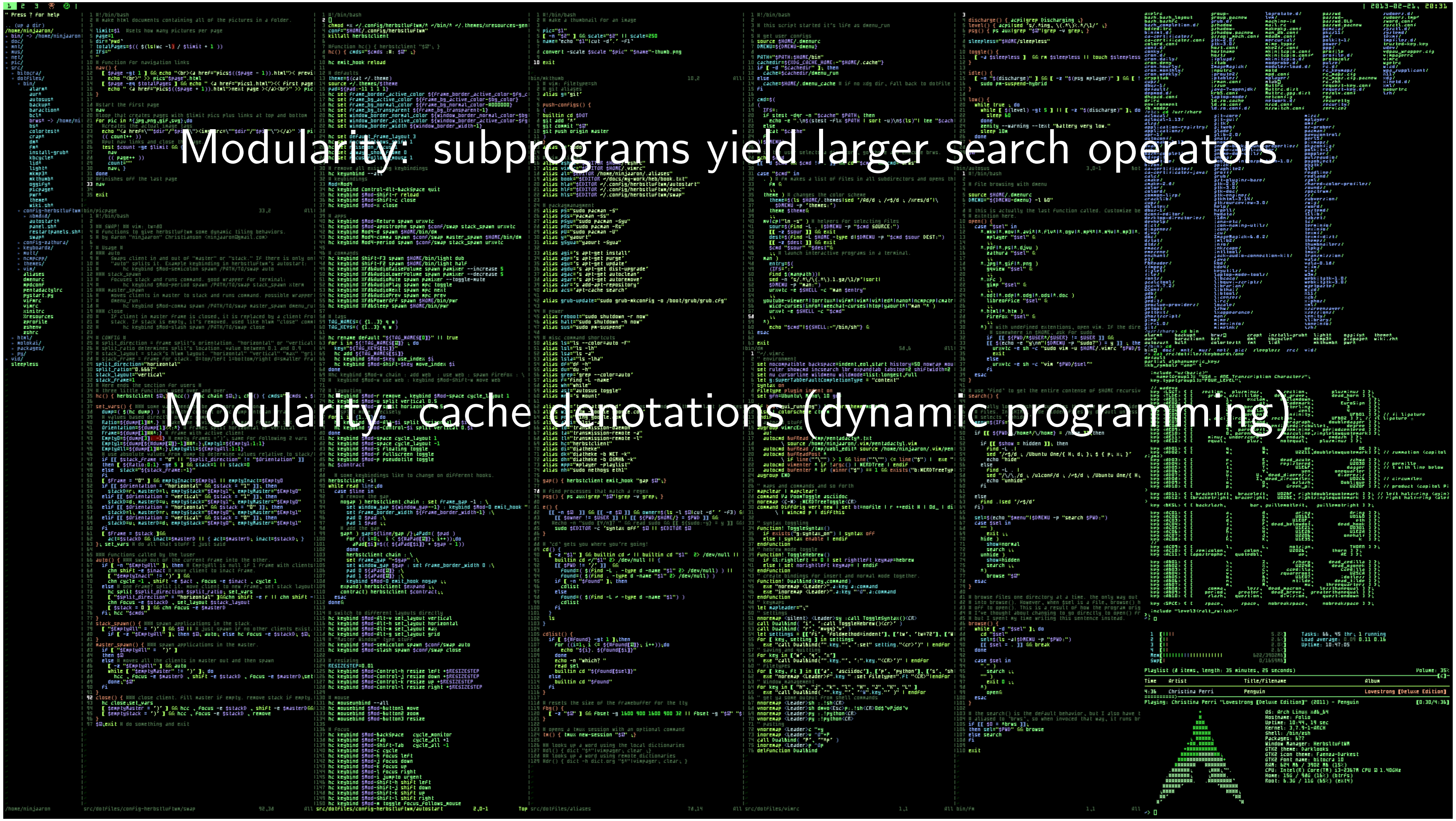
```
  1  [||||                          5.2%]     Tasks: 66, 45 thr; 1 running
  2  [||                            2.6%]     Load average: 0.09 0.11 0.16
  3  [||                            2.0%]     Uptime: 10:47:05
  4  [||                            2.6%]
  Mem[||||||||||||||||||||   622/3902MB]
  Swp[|                        0/1659MB]
```

```
Playlist (8 items, length: 35 minutes, 25 seconds)                         Volume: 35%
──────────────────────────────────────────────────────────────────────────────────[c]─
Time    Artist               Title/Filename                  Album
──────────────────────────────────────────────────────────────────────────────────────
4:36    Christina Perri      Penguin                         Lovestrong [Deluxe Edition]
==========>───────────────────────────────────────────────────────────────────────────
Playing: Christina Perri "Lovestrong [Deluxe Edition]" (2011) - Penguin      [0:30/4:36]
```

```
                 +                OS: Arch Linux x86_64
                 #                Hostname: folio
                ###               Uptime: 10:44, 19 sec
               #####              Kernel: 3.7.9-1-ARCH
               ######             Shell: /bin/zsh
              ; #####;            Packages: 677
             +##.#####            Window Manager: HerbstluftWM
            +##########           GTK2 theme: Darklooks
           #############;         GTK2 icon theme: Faenza-Darkest
          ###############+        GTK2 font name: bitocra 10
         #######   #######        RAM: 624 MB / 3902 MB (15%)
       .######;     ;###;`".      CPU: Intel(R) Core(TM) i3-2367M CPU @ 1.40GHz
      .#######;     ;#####.       Home: 15G / 98G (16%) (btrfs)
      #########.   .########`     Root: 6.3G / 11G (65%) (ext4)
     ######'           '######
    ;####                 ####;
    ##'                     '##
   #'                         `#
~> ▯
```

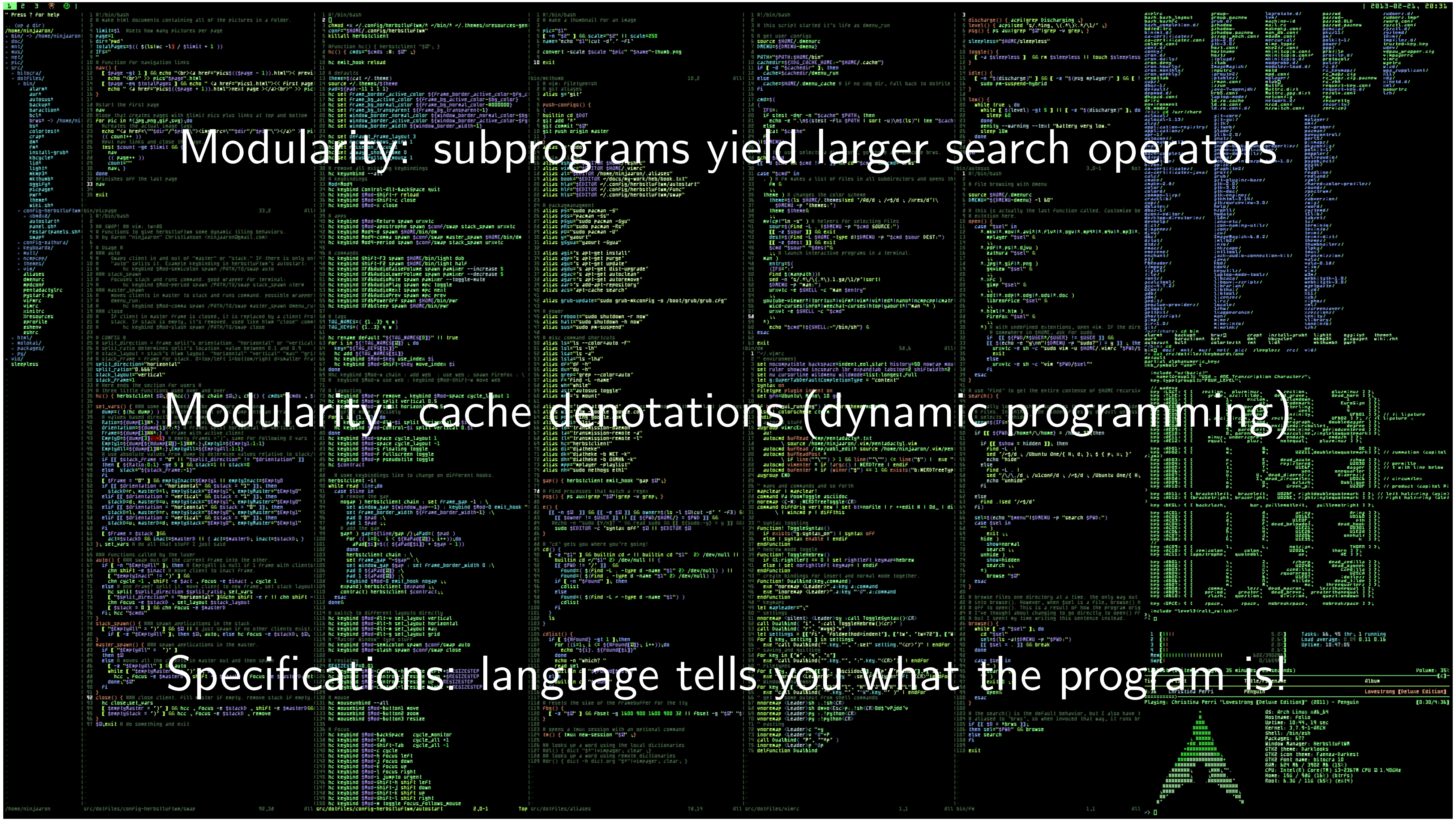
Modularity: subprograms yield larger search operators





Modularity: subprograms yield larger search operators

Modularity: cache denotations (dynamic programming)



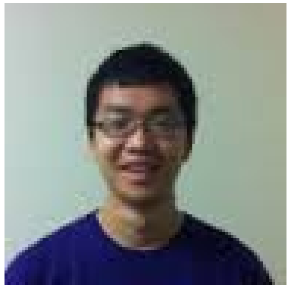
Modularity: subprograms yield larger search operators

Modularity: cache denotations (dynamic programming)

Specifications: language tells you what the program is!

Collaborators

Ice Pasupat


Robin Jia

Michael Jordan

Dan Klein

Code, data, experiments

CodaLab

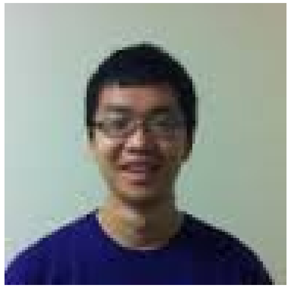
worksheets.codalab.org

Funding

Google Microsoft DARPA

Collaborators

Ice Pasupat


Robin Jia

Michael Jordan

Dan Klein

Code, data, experiments

CodaLab

worksheets.codalab.org

Funding

Google Microsoft DARPA

Thank you!