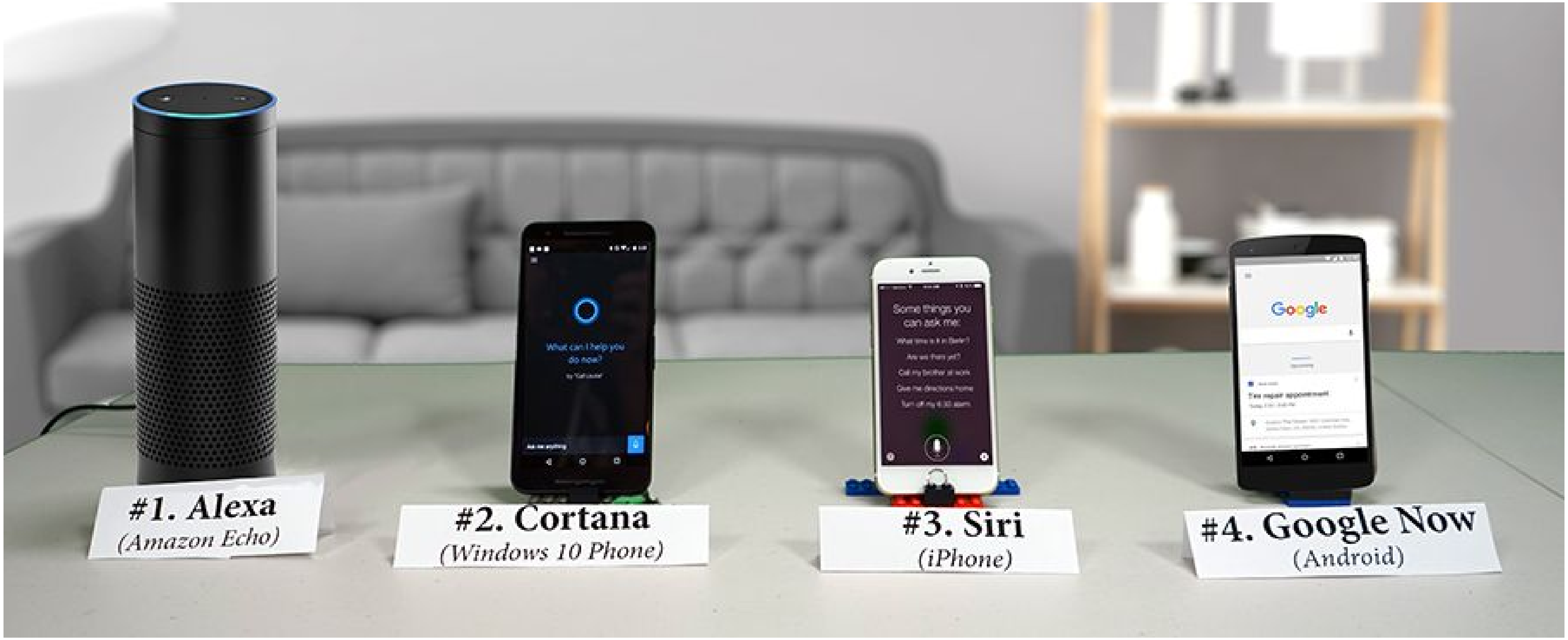
Naturalizing a programming language via interactive learning

Sida I. Wang, Samuel Ginn
Percy Liang, Christopher D. Manning

ACL 2017, Vancouver BC

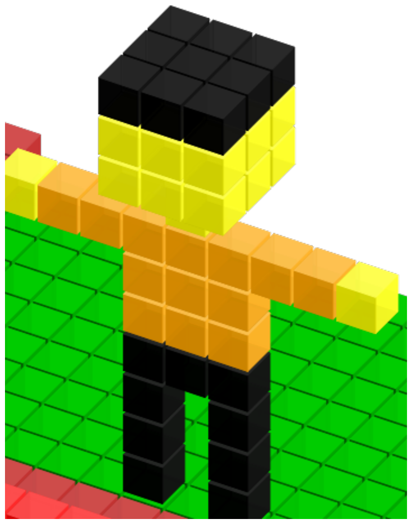
Natural language interfaces



how can they perform more complex actions?

Semantic parsing

natural language x to an executable program z

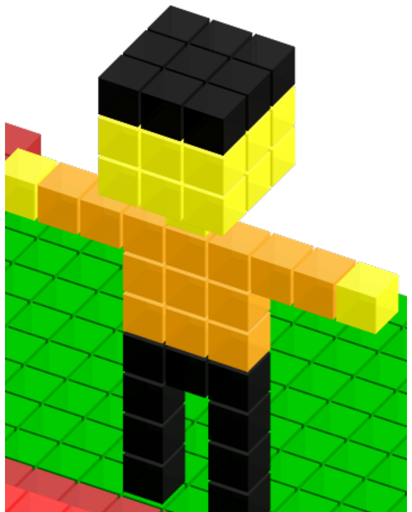


- human produces some utterance
- *add 3 red blocks on this*

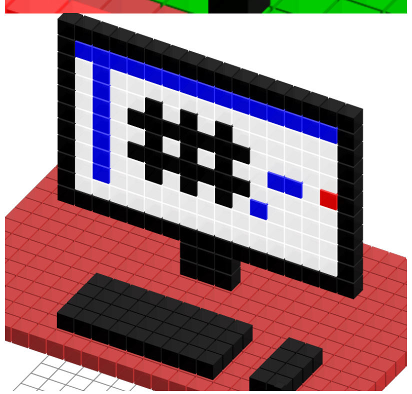
Zelle and Mooney 1996, Zettlemoyer and Collins 2005, Pasupat and Liang 2015

Semantic parsing

natural language x to an executable program z



- human produces some utterance
- *add 3 red blocks on this*

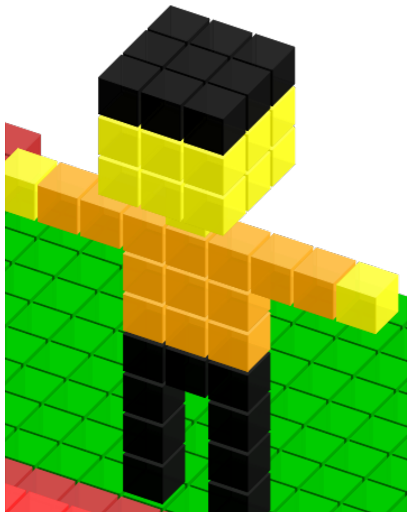


- converts utterance to a program

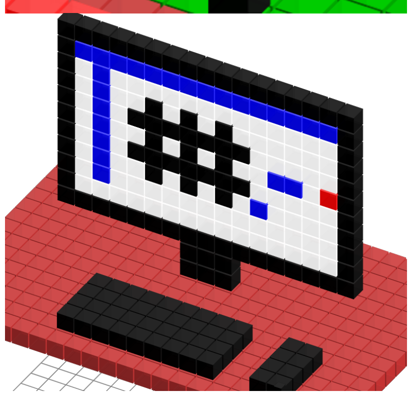
Zelle and Mooney 1996, Zettlemoyer and Collins 2005, Pasupat and Liang 2015

Semantic parsing

natural language x to an executable program z



- human produces some utterance
- *add 3 red blocks on this*

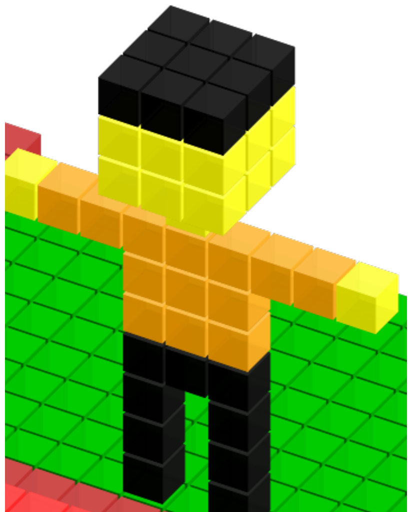


- converts utterance to a program
- `(loop 3 (add red top))`

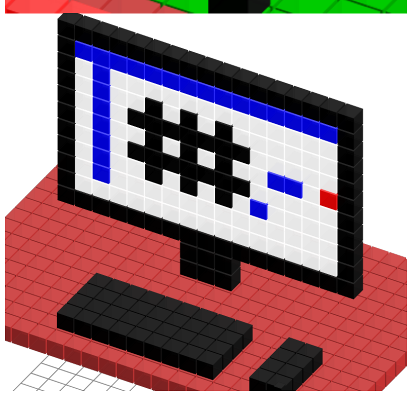
Zelle and Mooney 1996, Zettlemoyer and Collins 2005, Pasupat and Liang 2015

Semantic parsing

natural language x to an executable program z



- human produces some utterance
- *add 3 red blocks on this*

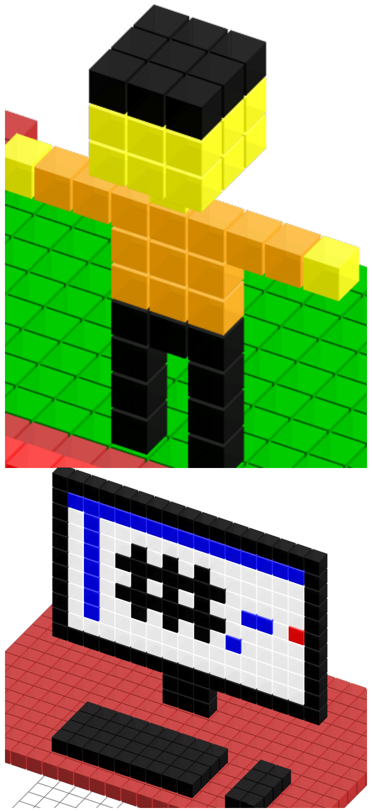


- converts utterance to a program
- `(loop 3 (add red top))`
- execute the program z
- produce a result y

Zelle and Mooney 1996, Zettlemoyer and Collins 2005, Pasupat and Liang 2015

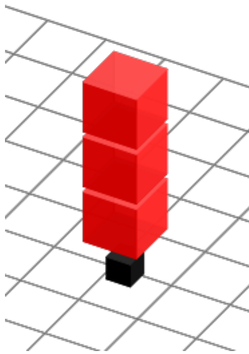
Semantic parsing

natural language x to an executable program z



- human produces some utterance
- *add 3 red blocks on this*

- converts utterance to a program
- `(loop 3 (add red top))`
- execute the program z
- produce a result y



Zelle and Mooney 1996, Zettlemoyer and Collins 2005, Pasupat and Liang 2015

Collect a static dataset

increase temperature by 3C

`setTemp(getTemp()+3)`

what is the largest state

`answer(A,largest(A,state(A)))`

people with children born in Vancouver

`Children.PlaceOfBirth.Vancouver`

add 3 red blocks on this

`(loop 3 (add red top))`

Collect a static dataset

increase temperature by 3C

`setTemp(getTemp()+3)`

what is the largest state

`answer(A,largest(A,state(A)))`

people with children born in Vancouver

`Children.PlaceOfBirth.Vancouver`

add 3 red blocks on this

`(loop 3 (add red top))`

then fit a model.

Collect a static dataset

increase temperature by 3C

```
setTemp(getTemp()+3)
```

what is the largest state

```
answer(A,largest(A,state(A)))
```

people with children born in Vancouver

```
Children.PlaceOfBirth.Vancouver
```

add 3 red blocks on this

```
(loop 3 (add red top))
```

then fit a model.

then deploy the system

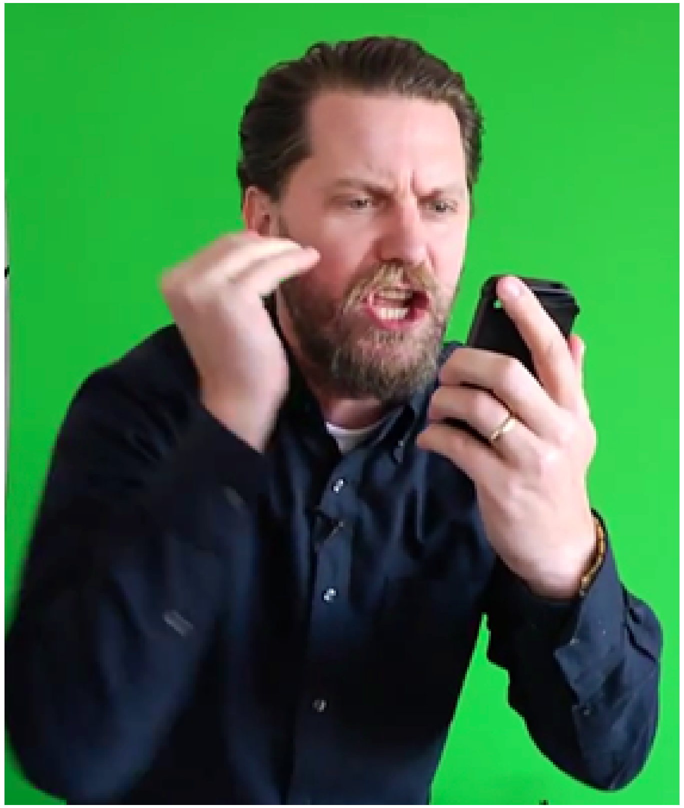
With a static system and static dataset

With a static system and static dataset

we are stuck when these systems misunderstand us

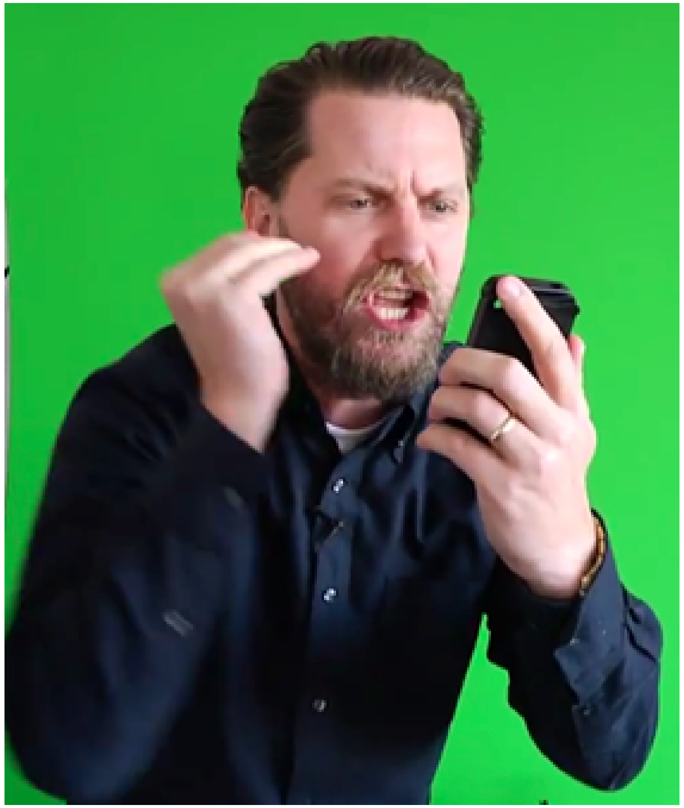
With a static system and static dataset

we are stuck when these systems misunderstand us



With a static system and static dataset

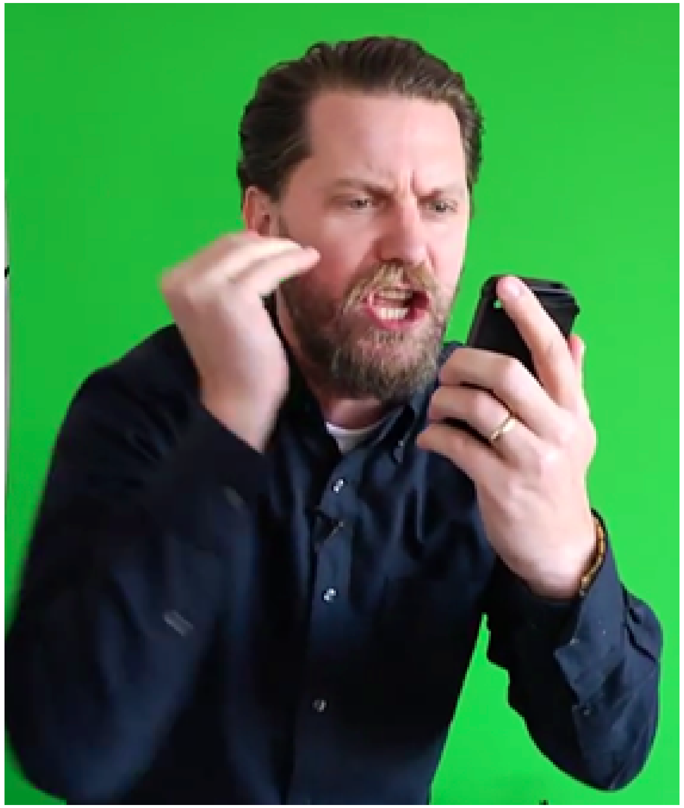
we are stuck when these systems misunderstand us



systems with interactive learning ability built-in

With a static system and static dataset

we are stuck when these systems misunderstand us

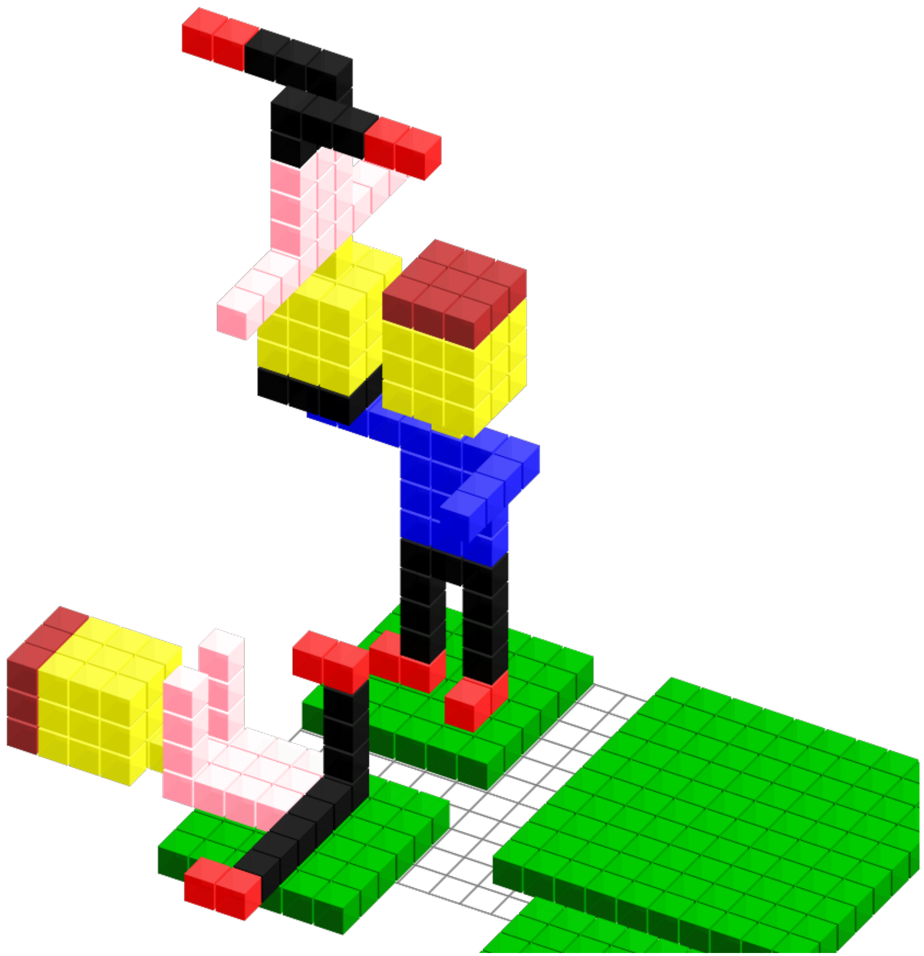


systems with interactive learning ability built-in

learn from users in real-time

Test ideas in blocks world

blocks world: intuitive, easy to crowdsource, and captures some unsolved problems



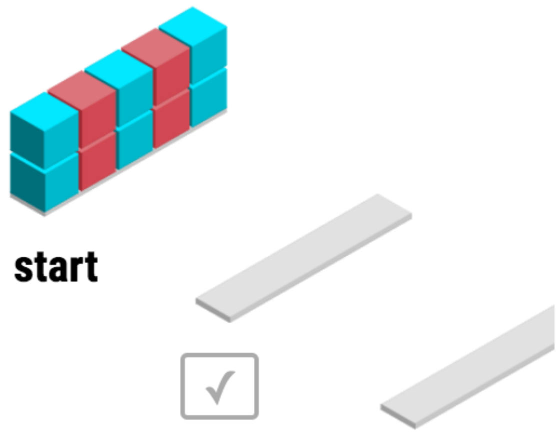
Interactive learning language games

Wittgenstein. 1953. Philosophical Investigations:

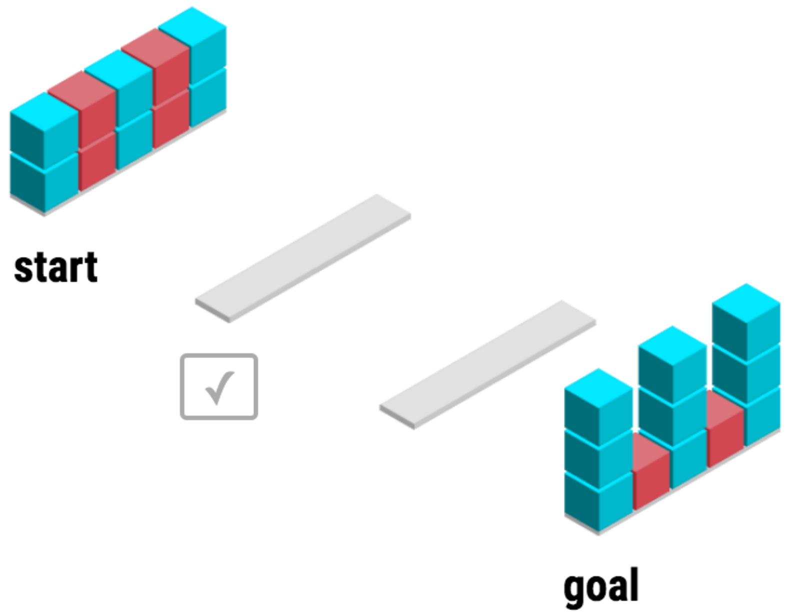
Language derives its meaning from use.

'block' 'pillar' 'slab' 'beam'.

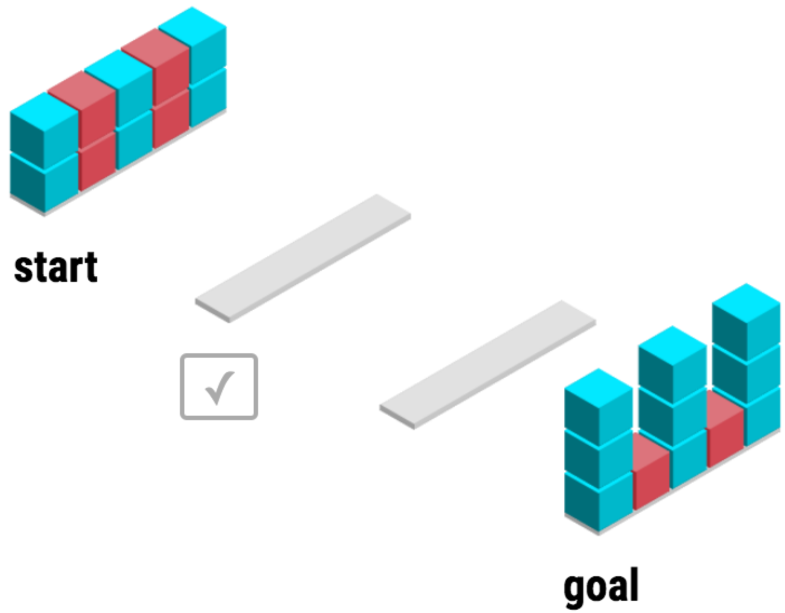
SHRDLURN

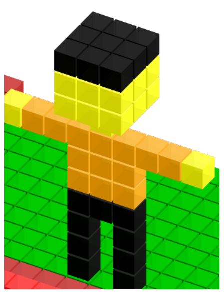


SHRDLURN

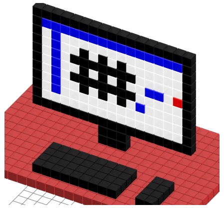


SHRDLURN





remove red



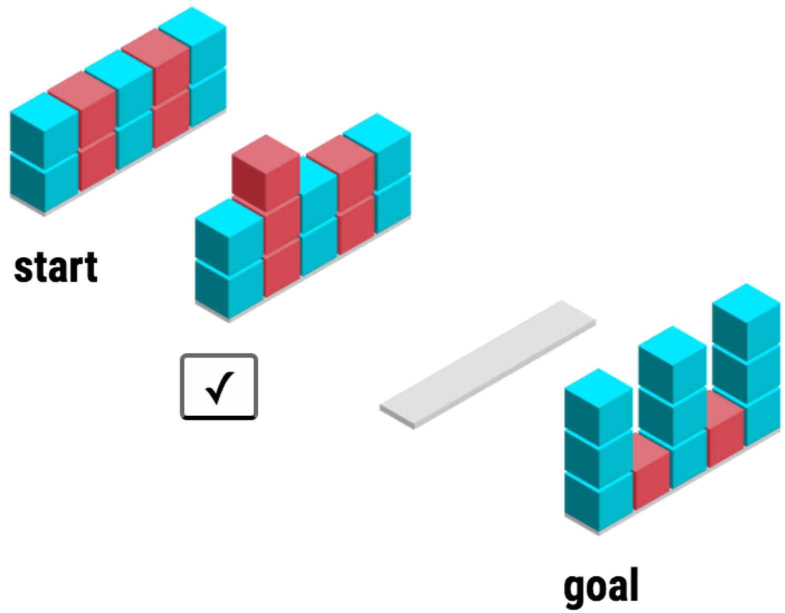
has a goal

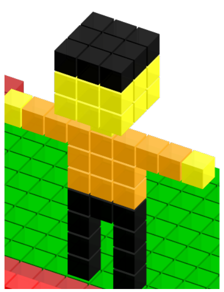
has language

performs actions

does not talk

SHRDLURN





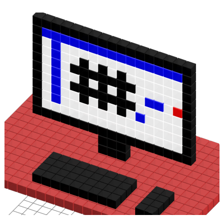
remove red

add(leftmost(hascolor(red)),red)

add(red, hascolor(cyan))

remove(hascolor(red))

remove(leftmost(hascolor(red)))



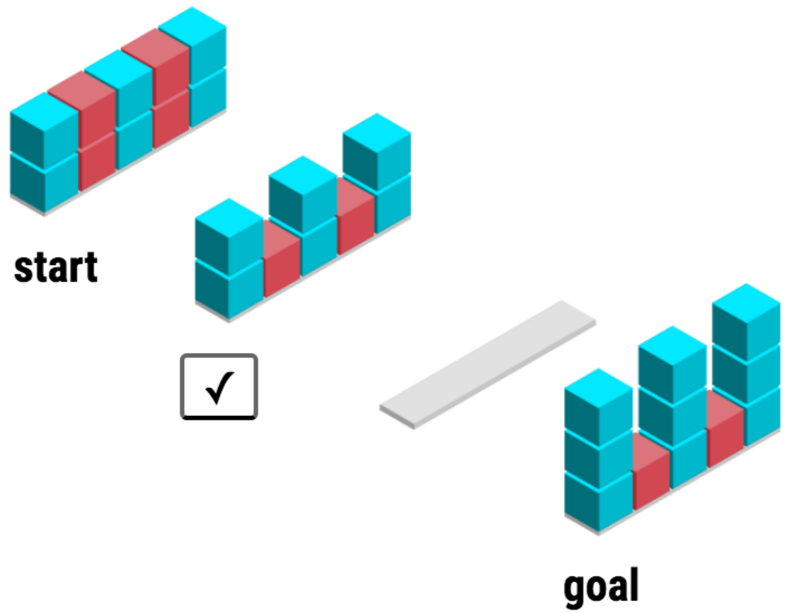
has a goal

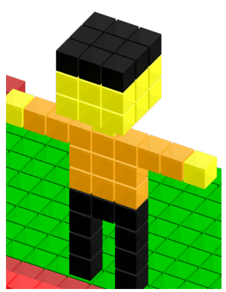
has language

performs actions

does not talk

SHRDLURN





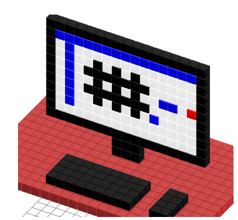
remove red

add(leftmost(hascolor(red)),red)

add(red, hascolor(cyan))

remove(hascolor(red))

remove(leftmost(hascolor(red)))



has a goal

has language

performs actions

does not talk

Learning signal is weak

- user provides a label via selection
- system can learn from denotation, logical form, or final reward

Learning signal is weak

- user provides a label via selection

- system can learn from denotation, logical form, or final reward

- for any learning to happen using these methods, the correct answer has to fall inside the search space

Learning signal is weak

- user provides a label via selection
- system can learn from denotation, logical form, or final reward
- for any learning to happen using these methods, the correct answer has to fall inside the search space
- works for short programs:
  - `answer(A,largest(A,state(A)))`
  - `remove(leftmost(hascolor(red)))`

Learning signal is weak

- user provides a label via selection

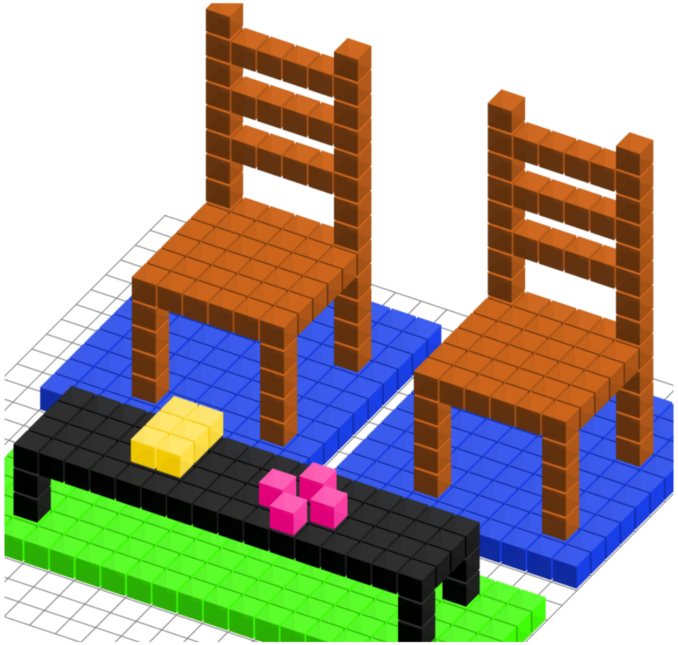
- system can learn from denotation, logical form, or final reward

- for any learning to happen using these methods, the correct answer has to fall inside the search space

- works for short programs:
  - `answer(A,largest(A,state(A)))`
  - `remove(leftmost(hascolor(red)))`
- cannot possibly scale to more complex programs

Cannot possibly keep up

legs of height 3 with 3 spaces apart

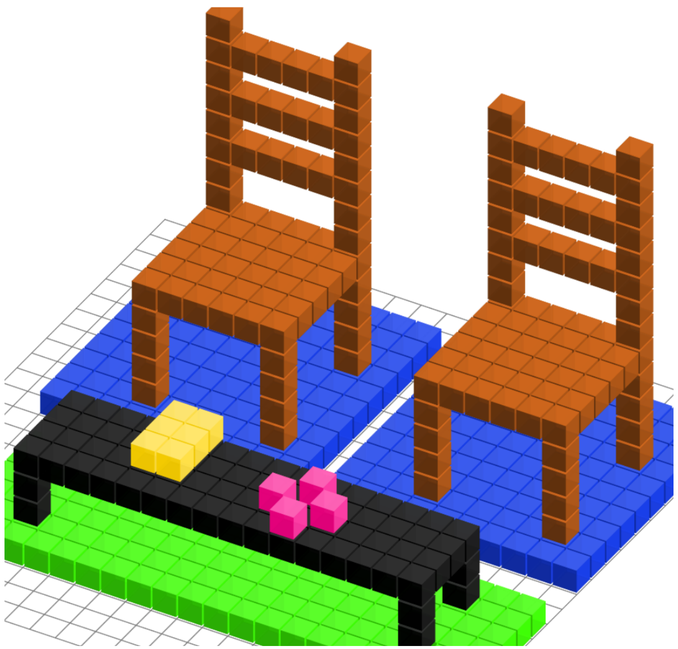


```
(:s (:s (:blkr (:s (:loop (number 3) (:s (:  add brown
here) (:for (call adj top this) (:  select)))) (:loop
(number 3) (:for (call adj bot this) (:  select))))
(:loop (number 3) (:for (call adj left this) (:
select)))) (:s (:s (:s (:s (:blkr (:s (:loop (number
3) (:s (:  add brown here) (:for (call adj top this)
(:  select)))) (:loop (number 3) (:for (call adj bot
this) (:  select))))) (:loop (number 3) (:for (call
adj back this) (:  select)))) (:blkr (:s (:loop (number
3) (:s (:  add brown here) (:for (call adj top this)
(:  select)))) (:loop (number 3) (:for (call adj bot
this) (:  select)))))) (:loop (number 3) (:for (call
adj right this) (:  select)))) (:blkr (:s (:loop (number
3) (:s (:  add brown here) (:for (call adj top this) (:
select)))) (:loop (number 3) (:for (call adj bot this)
(:  select)))))))
```

number of programs of this length $> 10^{100}$

Cannot possibly keep up

legs of height 3 with 3 spaces apart



```
(:s (:s (:blkr (:s (:loop (number 3) (:s (:  add brown
here) (:for (call adj top this) (:  select)))) (:loop
(number 3) (:for (call adj bot this) (:  select)))))
(:loop (number 3) (:for (call adj left this) (:
select)))) (:s (:s (:s (:s (:blkr (:s (:loop (number
3) (:s (:  add brown here) (:for (call adj top this)
(:  select)))) (:loop (number 3) (:for (call adj bot
this) (:  select))))) (:loop (number 3) (:for (call
adj back this) (:  select)))) (:blkr (:s (:loop (number
3) (:s (:  add brown here) (:for (call adj top this)
(:  select)))) (:loop (number 3) (:for (call adj bot
this) (:  select)))))) (:loop (number 3) (:for (call
adj right this) (:  select)))) (:blkr (:s (:loop (number
3) (:s (:  add brown here) (:for (call adj top this) (:
select)))) (:loop (number 3) (:for (call adj bot this)
(:  select)))))))
```

number of programs of this length $> 10^{100}$

need stronger supervision to produce such programs

demonstrations, instructions, definitions

Naturalizing a programming language

start with a core programming language

- starting point of definitions

Naturalizing a programming language

start with a core programming language

- starting point of definitions

a community of users interact with the system

- more definition using core and previous definitions
- leg of height 3 $:=$ brown column of height 3

Naturalizing a programming language

start with a core programming language

- starting point of definitions

a community of users interact with the system

- more definition using core and previous definitions
- leg of height 3 := brown column of height 3

induce grammar rule

Naturalizing a programming language

start with a core programming language

- starting point of definitions

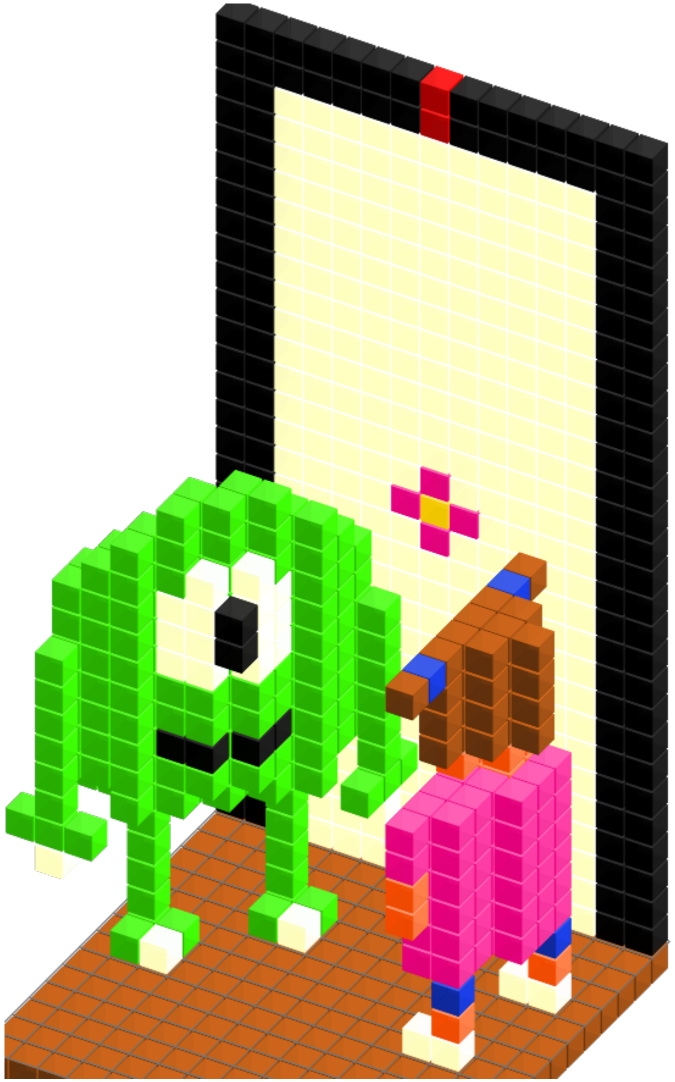
a community of users interact with the system

- more definition using core and previous definitions
- leg of height 3 := brown column of height 3

induce grammar rule

- "learning macros from crowd programming"
- no explicit arguments and variables

Voxelurn



world is a set of objects

with relations

voxels: (x, y, z, color)

relations: left, top, front, etc.

actions: select, add, move

Core language

- controls: if, foreach, repeat, while
- block-structured scoping
  - , [], isolate

Core language

- controls: if, foreach, repeat, while
- block-structured scoping
    - , [], isolate
- lambda DCS for sets

yellow blocks in row 1

λ-DCS: has color yellow and has row 1

Core language

- controls: if, foreach, repeat, while
- block-structured scoping
  - , [], isolate
- lambda DCS for sets

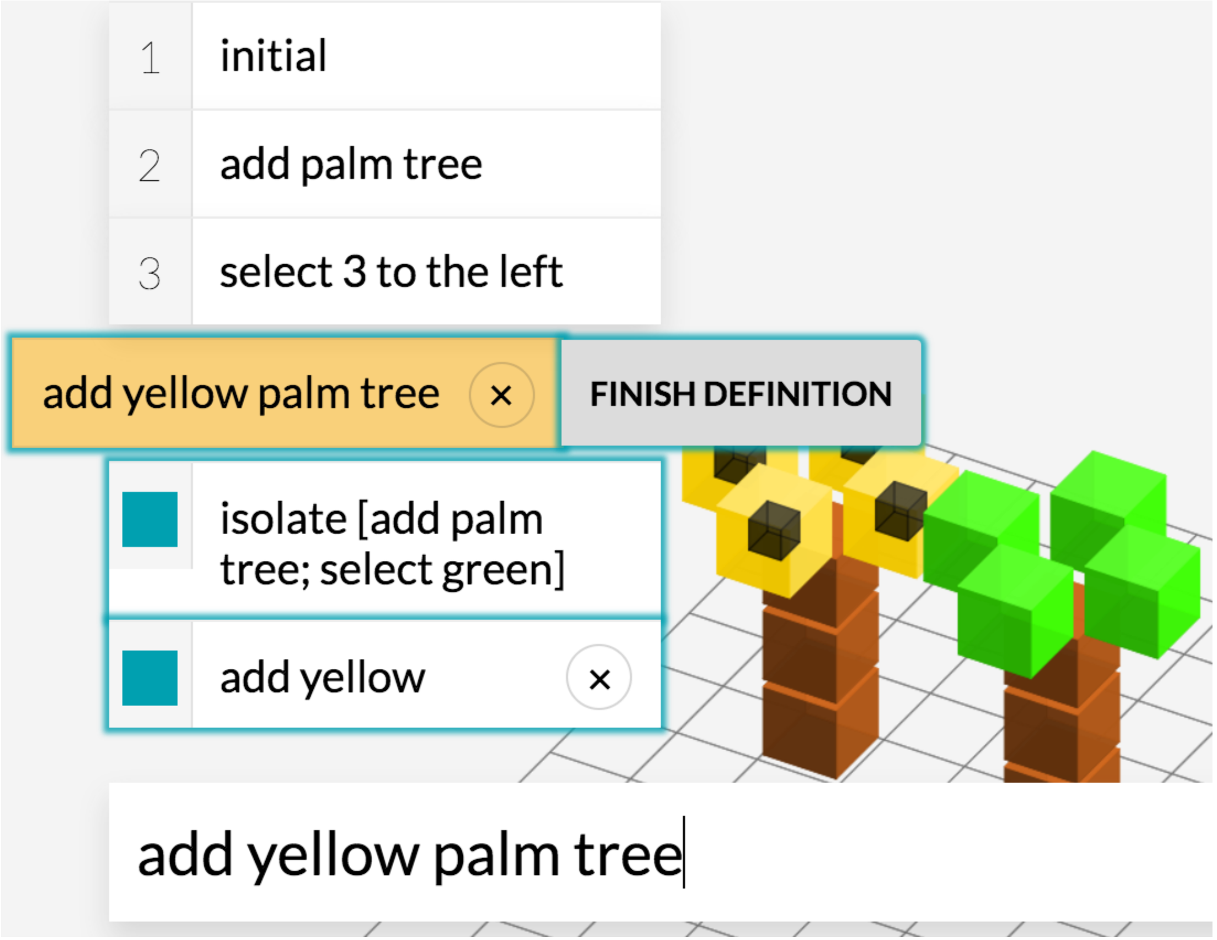
yellow blocks in row 1

λ-DCS: has color yellow and has row 1

- selection as the default argument
  - add red top (to selected)

Lets make some trees

- define new things in terms of what's already defined
- trace back to the core language



Palm tree example

define new concepts in terms of what's already defined

everything trace back to the core language

add palm tree:

    add brown trunk height 3:

    go to top:

    add leaves here:

Palm tree example

define new concepts in terms of what's already defined

everything trace back to the core language

add palm tree:

    add brown trunk height 3:

        add brown top 3 times:

    go to top:

    add leaves here:

Palm tree example

define new concepts in terms of what's already defined

everything trace back to the core language

```
add palm tree:
    add brown trunk height 3:
        add brown top 3 times:
            repeat 3 [add brown top]
    go to top:


    add leaves here:
```

Palm tree example

define new concepts in terms of what's already defined

everything trace back to the core language

```
add palm tree:
    add brown trunk height 3:
        add brown top 3 times:
            repeat 3 [add brown top]
    go to top:
        select very top of all
    add leaves here:
```

Palm tree example

define new concepts in terms of what's already defined

everything trace back to the core language

add palm tree:

    add brown trunk height 3:

        add brown top 3 times:

            repeat 3 [add brown top]

    go to top:

        select very top of all

    add leaves here:

        select left or right or front or back; add green

Inside a definition

unparsable head: *add red left 3 times*

Inside a definition

unparsable head: *add red left 3 times*

parsable body: *repeat 3 add red left*

derivation of the body: `(loop 3 (add red left))`

Inside a definition

unparsable head: *add red left 3 times*

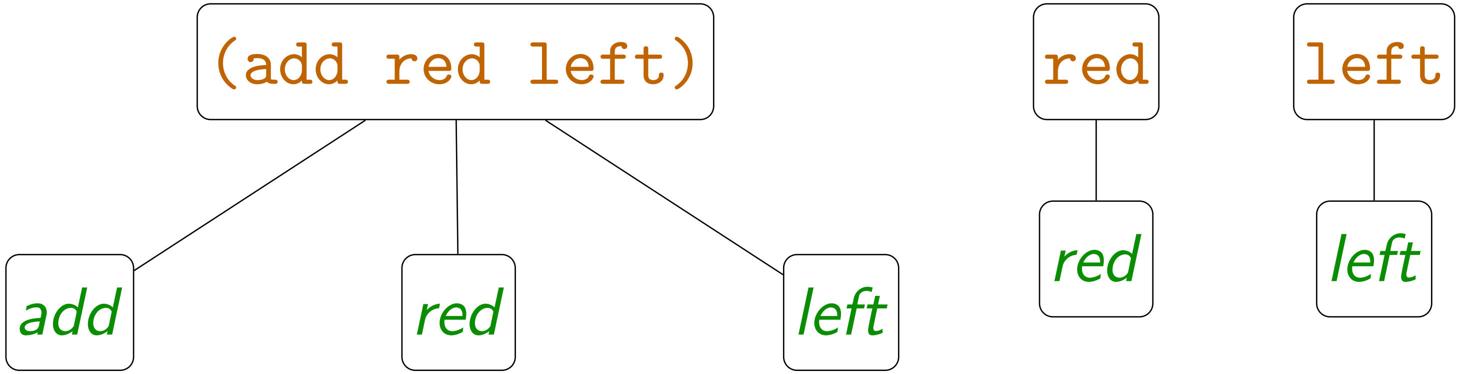
parsable parts of the head: *red, left, 3, add red left*

parsable body: *repeat 3 add red left*

derivation of the body: `(loop 3 (add red left))`

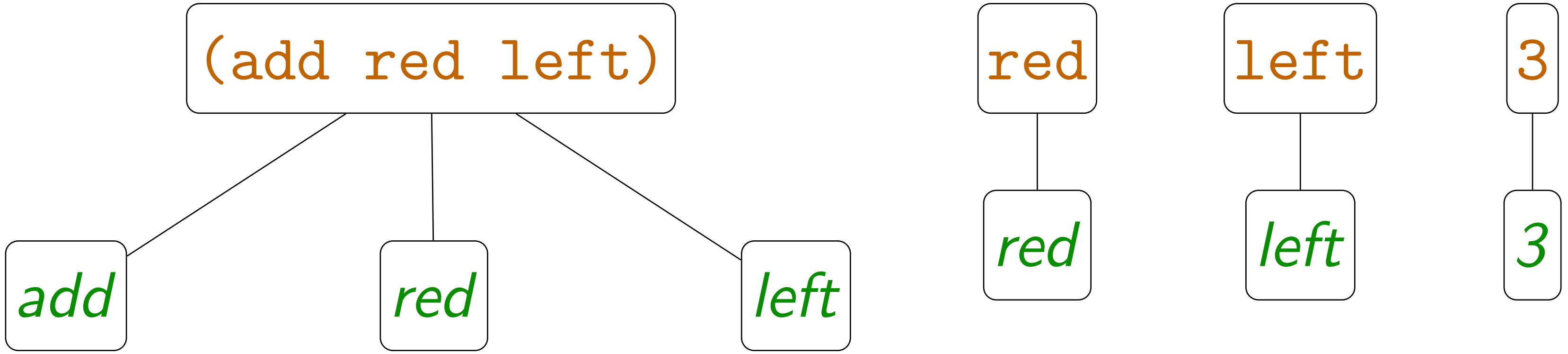
Unparsable head

unparsable head: *add red left 3 times*



Unparsable head

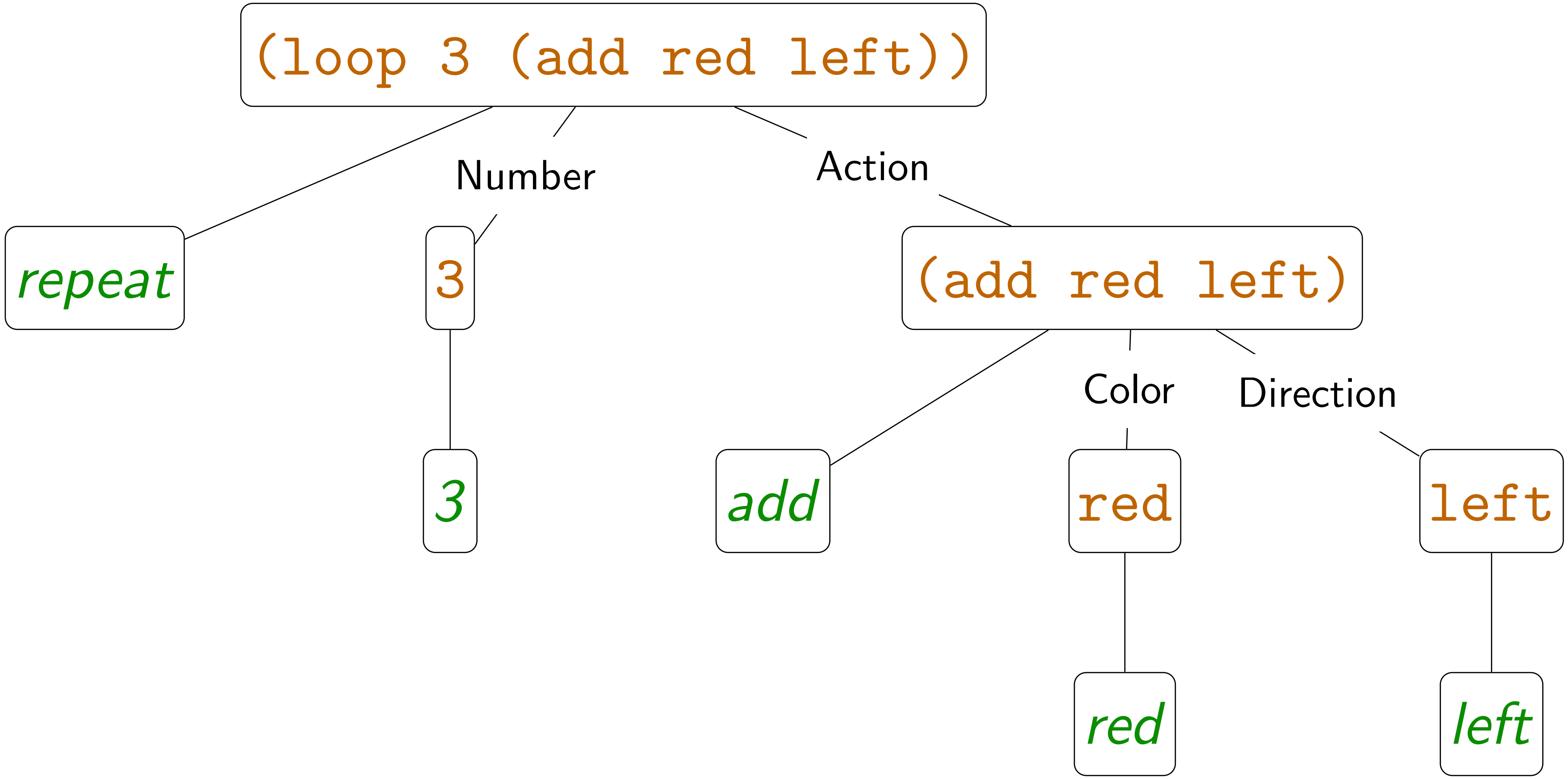
unparsable head: *add red left 3 times*



Parsable body

unparsable head: *add red left 3 times*

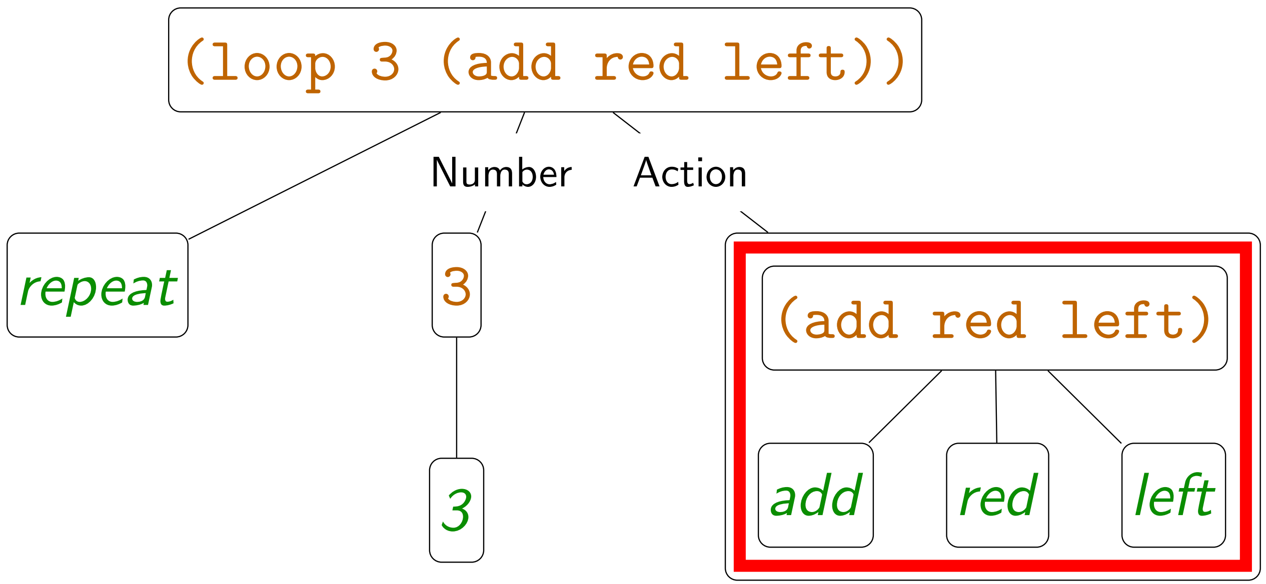
parsable body: *repeat 3 add red left*



Substitute matches

unparsable head: add red left 3 times

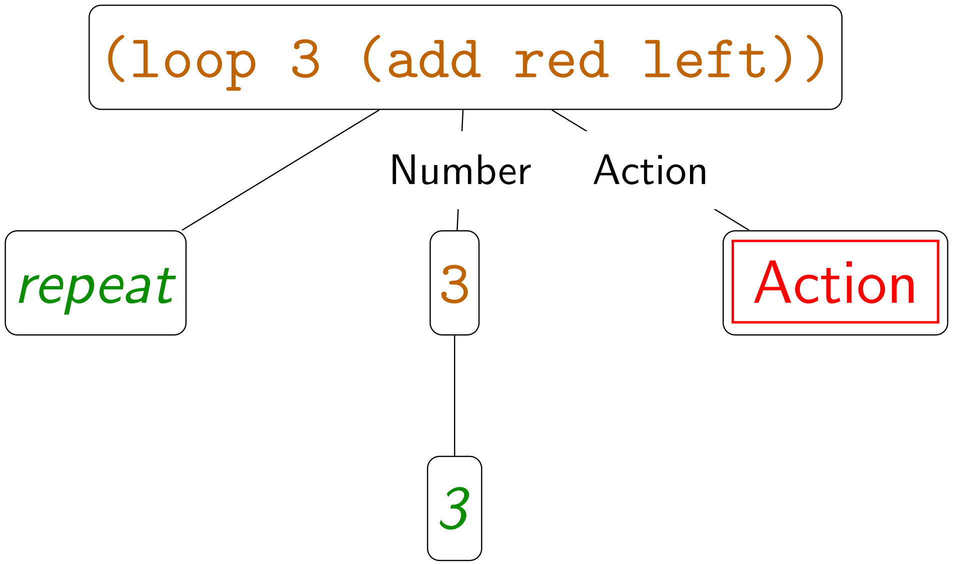
parsable body: repeat 3 add red left



Substitute matches

unparsable head: add red left 3 times

parsable body: repeat 3 add red left

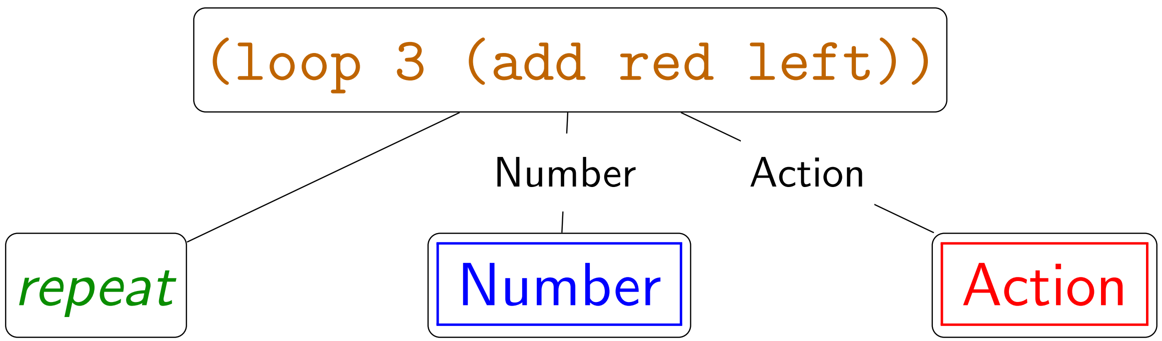


induced rule: $A \rightarrow A \;\; 3 \;\;$ times $\; : \lambda A.$(loop 3 A)

Substitute matches

unparsable head: add red left 3 times

parsable body: repeat 3 add red left



induced rule: $A \rightarrow A \quad N$ times $: \lambda A\ N.$(loop N A)

Matches not unique

more abstract

$\boxed{\text{add red left}}$ $\boxed{3}$ times

repeat $\boxed{3}$ $\boxed{\text{add red left}}$

$A \rightarrow A \quad N \quad \text{times} \; : \lambda A \; N.\text{(loop N A)}$

less abstract

add $\boxed{\text{red}}$ $\boxed{\text{left}}$ $\boxed{3}$ times

repeat $\boxed{3}$ add $\boxed{\text{red}}$ $\boxed{\text{left}}$

$A \rightarrow \text{add} \; C \; D \; N \; \text{times} \; : \lambda C \; D \; N.\text{(loop N (add C D))}$

Take highest scoring ones

- a packing is a set of non-overlapping potential matches
  - maximal packing – no span can be added

$\boxed{\text{add red left}}$ $\boxed{3}$ times

repeat $\boxed{3}$ $\boxed{\text{add red left}}$

Take highest scoring ones

- a packing is a set of non-overlapping potential matches
  - maximal packing – no span can be added

add $\boxed{\text{red}}$ $\boxed{\text{left}}$ $\boxed{3}$ times

repeat $\boxed{3}$ add $\boxed{\text{red}}$ $\boxed{\text{left}}$

Zettlemoyer and Collins 2005, 2007, Kwiatkowski et al., 2010

Take highest scoring ones

- a packing is a set of non-overlapping potential matches
  - maximal packing – no span can be added

add $\boxed{\text{red}}$ $\boxed{\text{left}}$ $\boxed{3}$ times

repeat $\boxed{3}$ add $\boxed{\text{red}}$ $\boxed{\text{left}}$

- abstract away the highest scoring maximal packing

$$P_l^* = \operatorname*{argmax}_{P \in \text{packing}(M);} \sum_{d \in P} \text{score}(d).$$

- solve with a dynamic program

Zettlemoyer and Collins 2005, 2007, Kwiatkowski et al., 2010

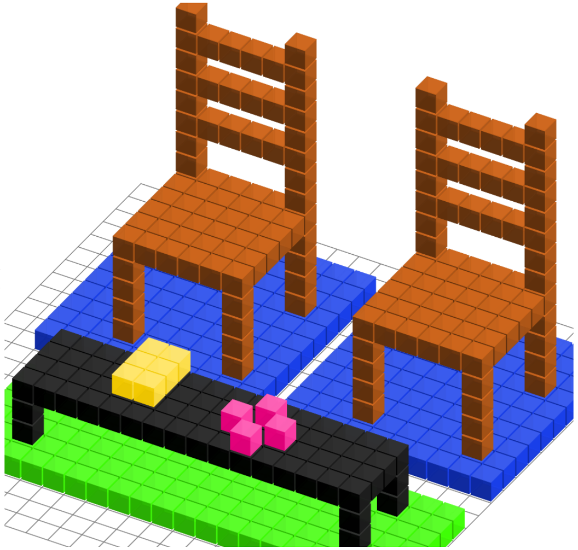
Model over derivations

log-linear model with features $\phi(d, x, u)$:

$$p_\theta(d \mid x, u) \propto \exp(\phi(d, x, u) \cdot \theta)$$

x : *add two chairs 5 spaces apart*

$z = \mathrm{formula}(d)$: (:blk (:loop ...))

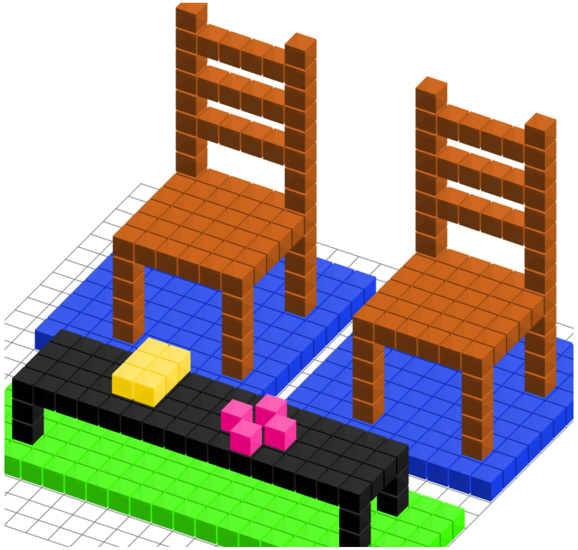
y : 

Learning from denotations

$$p_\theta(d \mid x, u) \propto \exp(\phi(d, x, u) \cdot \theta)$$

x : *add two chairs 5 spaces apart*

$z = \text{formula}(d)$: (:blk (:loop ...))

y : 

Learning from denotations

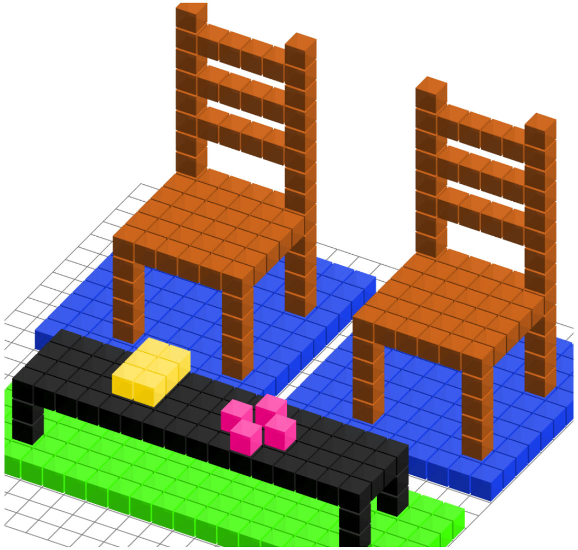
$$p_\theta(d \mid x, u) \propto \exp(\phi(d, x, u) \cdot \theta)$$

$$p_\theta(y \mid x, u) = \sum_{d:\text{Exec}(d)=y} p_\theta(d \mid x, y)$$

x : *add two chairs 5 spaces apart*

$z = \text{formula}(d) : (: blk\ (: loop...))$

y :



Learning from denotations

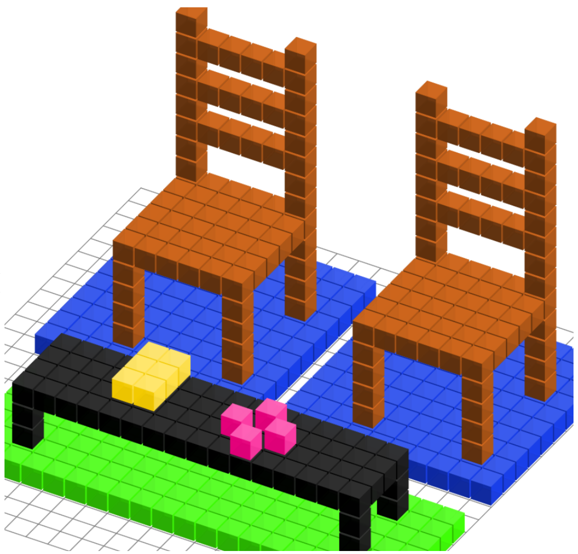
$$p_\theta(d \mid x, u) \propto \exp(\phi(d, x, u) \cdot \theta)$$

$$p_\theta(y \mid x, u) = \sum_{d:\mathrm{Exec}(d)=y} p_\theta(d \mid x, y)$$

x : *add two chairs 5 spaces apart*

$z = \mathrm{formula}(d)$: $(: blk\ (: loop ...))$

y :



L1 penalty and update with AdaGrad

Features

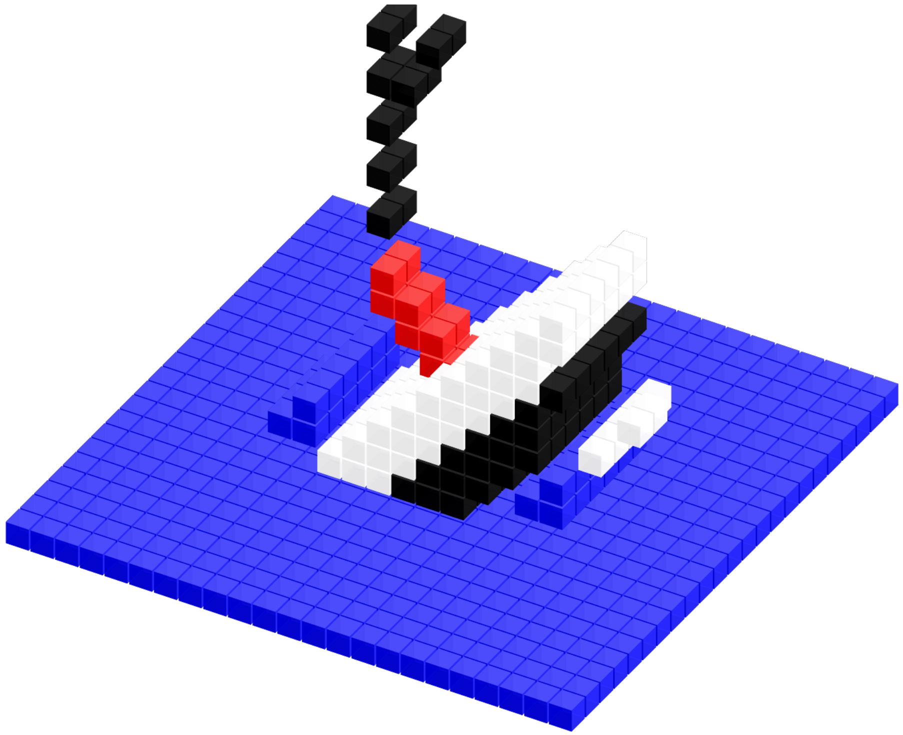
- generic: ruleId, span
- rule type: core? induced? used?
- social: authorId, (authorId, userId), self?
  - captures user community

Experiments

can crowdworkers provide such strong supervision?

initial users have to learn the core language

following user can build on previous users

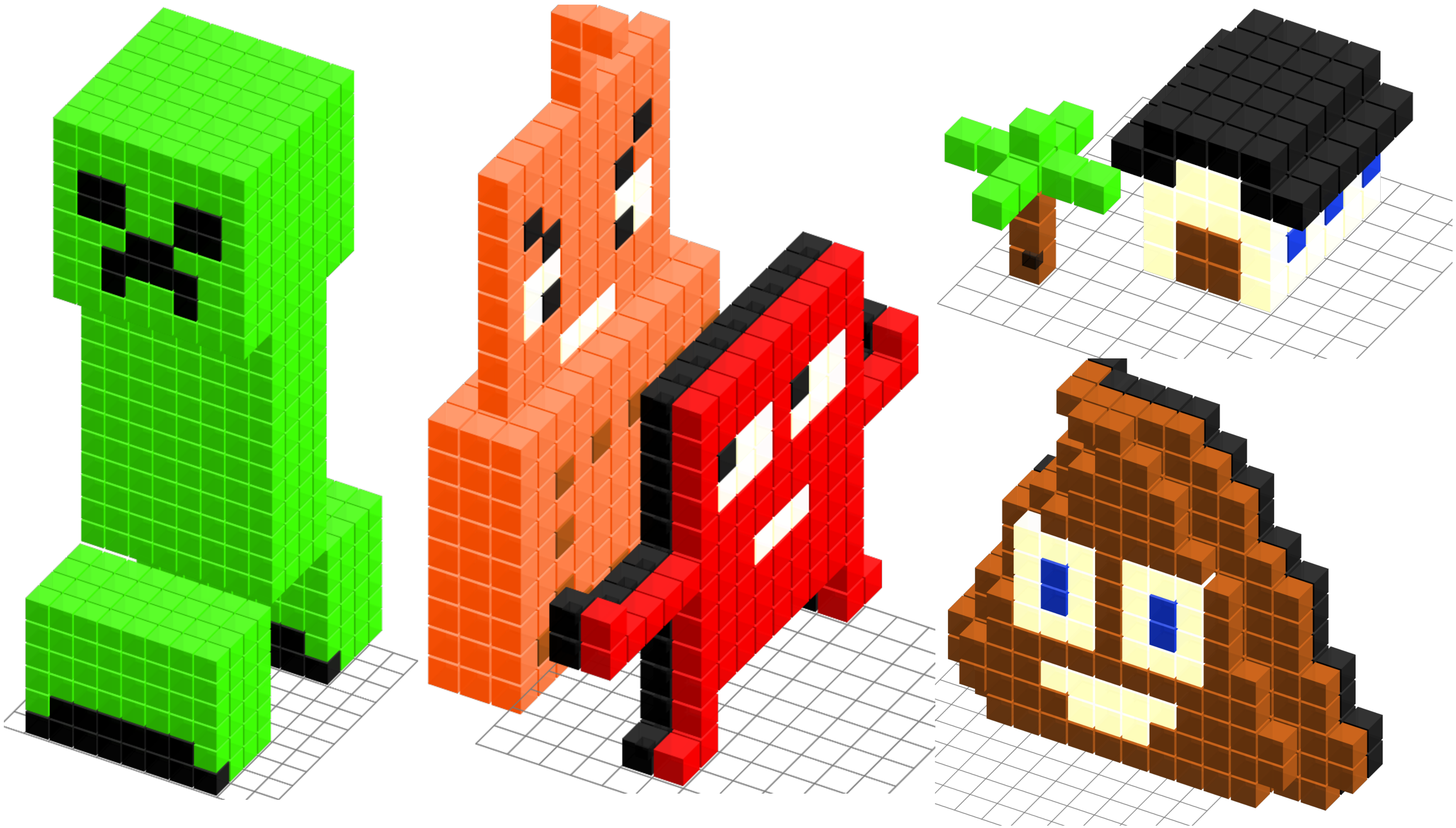


chair legs of height 3

```
(:s (:s (:blkr (:s (:loop (number 3) (:s (:  add brown
here) (:for (call adj top this) (:  select)))) (:loop
(number 3) (:for (call adj bot this) (:  select)))))
(:loop (number 3) (:for (call adj left this) (:
select)))) (:s (:s (:s (:s (:blkr (:s (:loop (number
3) (:s (:  add brown here) (:for (call adj top this)
(:  select)))) (:loop (number 3) (:for (call adj bot
this) (:  select))))) (:loop (number 3) (:for (call
adj back this) (:  select)))) (:blkr (:s (:loop (number
3) (:s (:  add brown here) (:for (call adj top this)
(:  select)))) (:loop (number 3) (:for (call adj bot
this) (:  select)))))) (:loop (number 3) (:for (call
adj right this) (:  select)))) (:blkr (:s (:loop (number
3) (:s (:  add brown here) (:for (call adj top this) (:
select)))) (:loop (number 3) (:for (call adj bot this)
(:  select)))))))
```

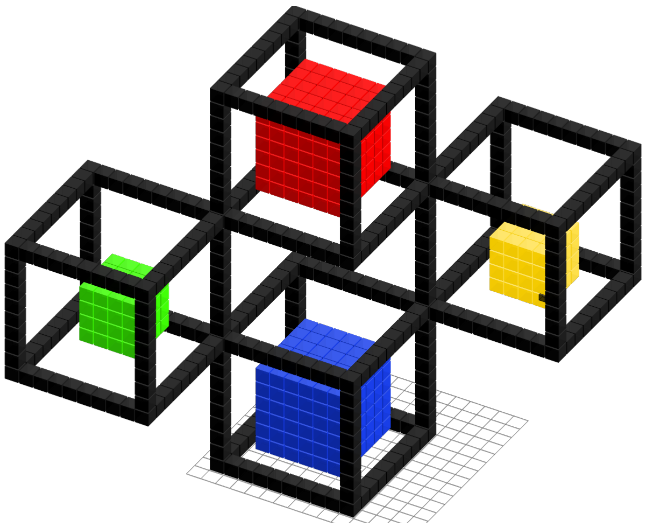
Experiments

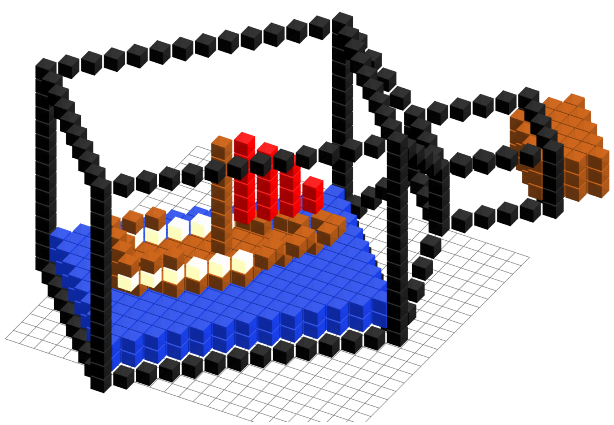
- users built great structures?

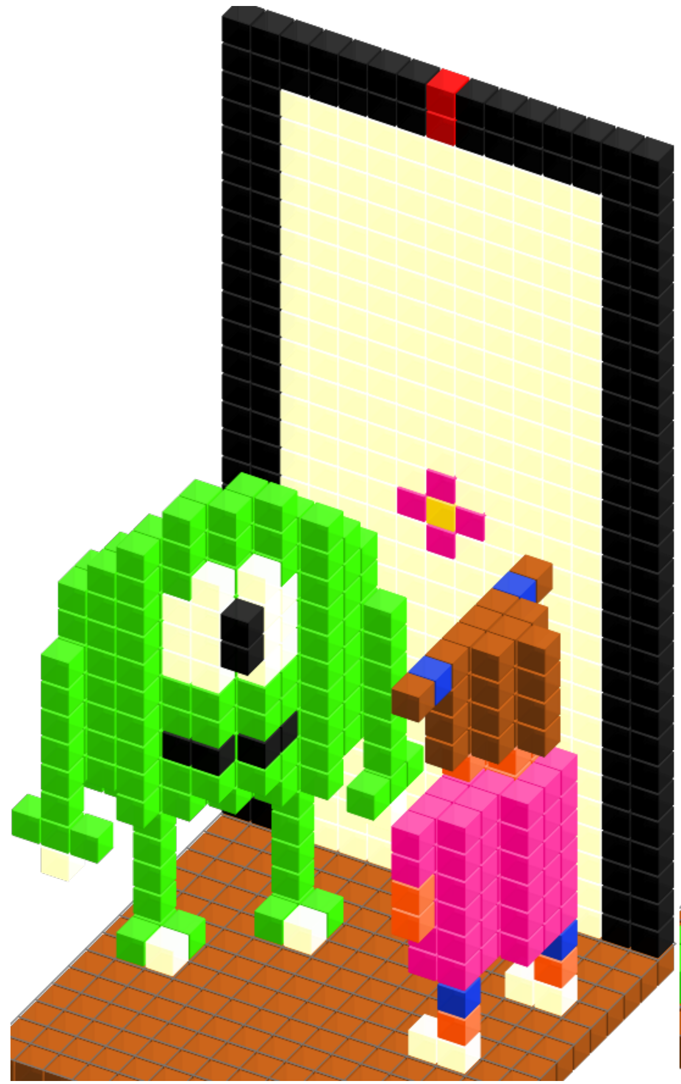


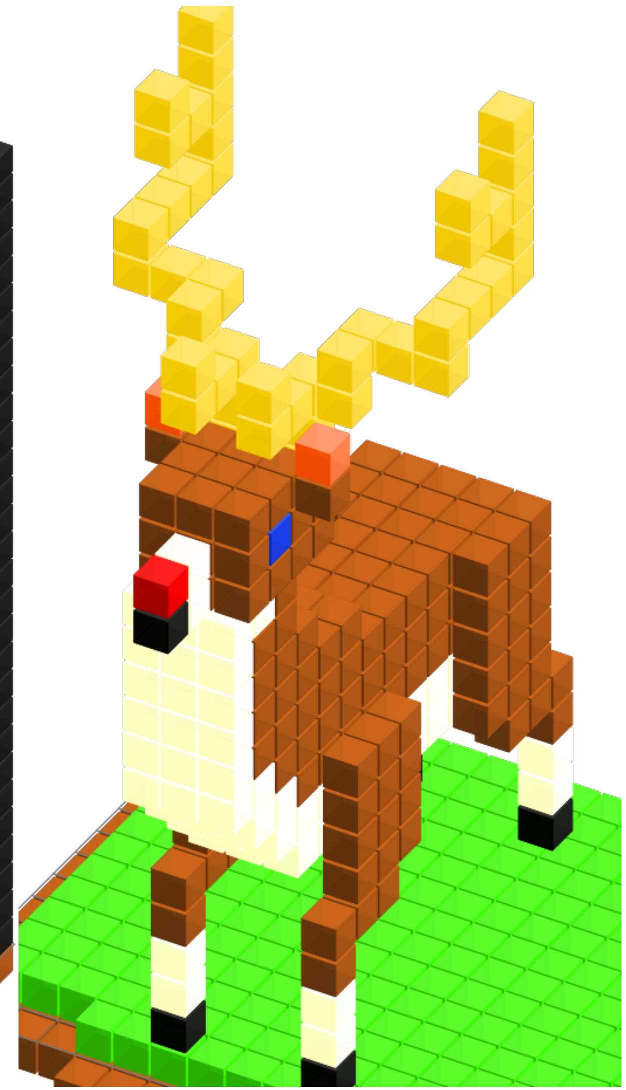
Experiments

- users built great structures! (show leaderboard)

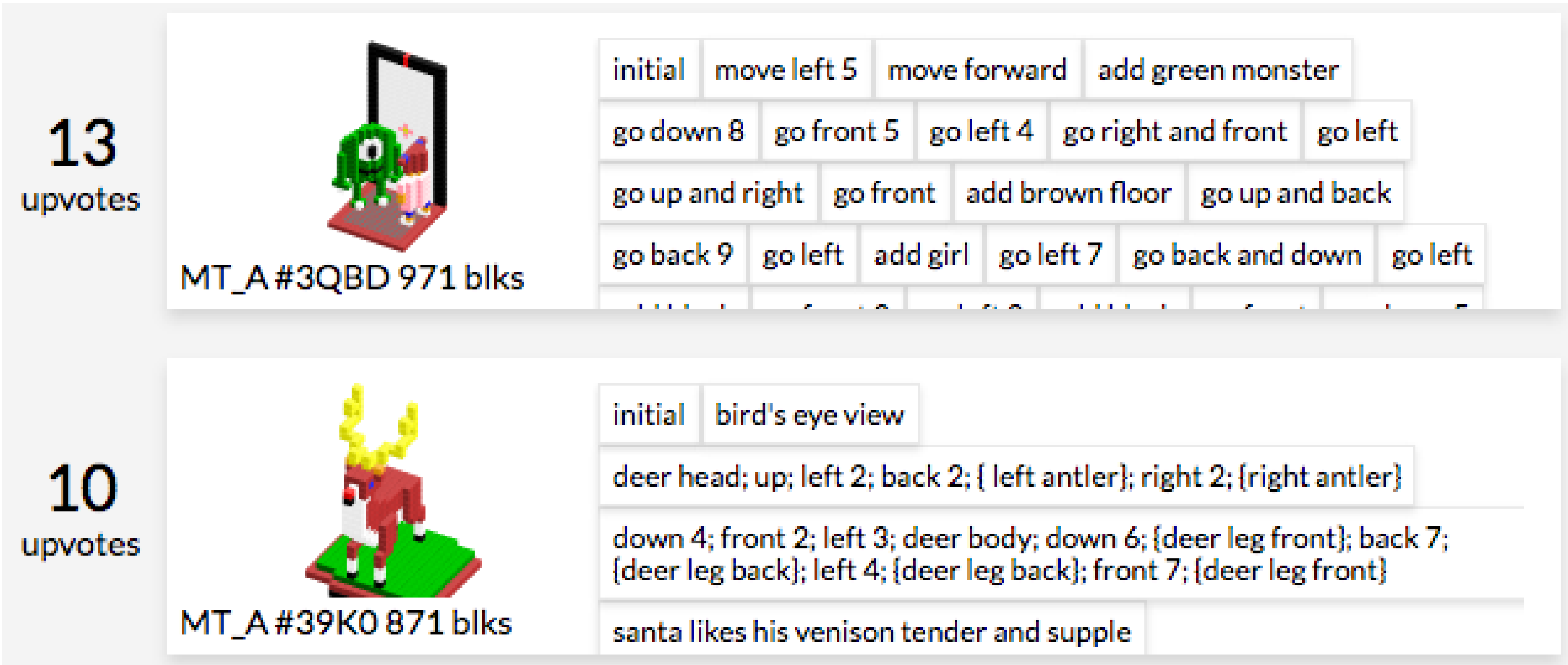








Leaderboard

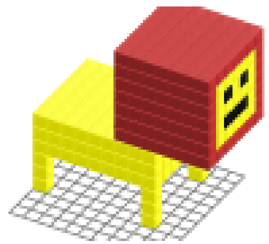


go down 5, go up and back, left 2

add green monster, deer head

Leaderboard

8 upvotes



MT_A #37QW 391 blks

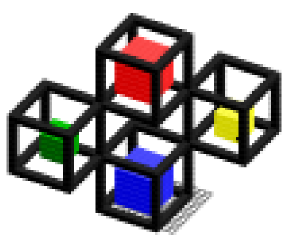
initial | go left 5 | go front 2 | add yellow column 3
add yellow platform | go back and right | add yellow front 6
add yellow | go right | add yellow front 6 | go down 3
add yellow column 3 | go right | go down 3 | go front 6 | go left

7 upvotes

MT_A #3X87 993 blks

initial | black bottom
barrier blue draw right; barrier blue draw back
right 6; back 3; up 3; power pellet
back 3; repeat 2 [right 6]; barrier blue point front

7 upvotes



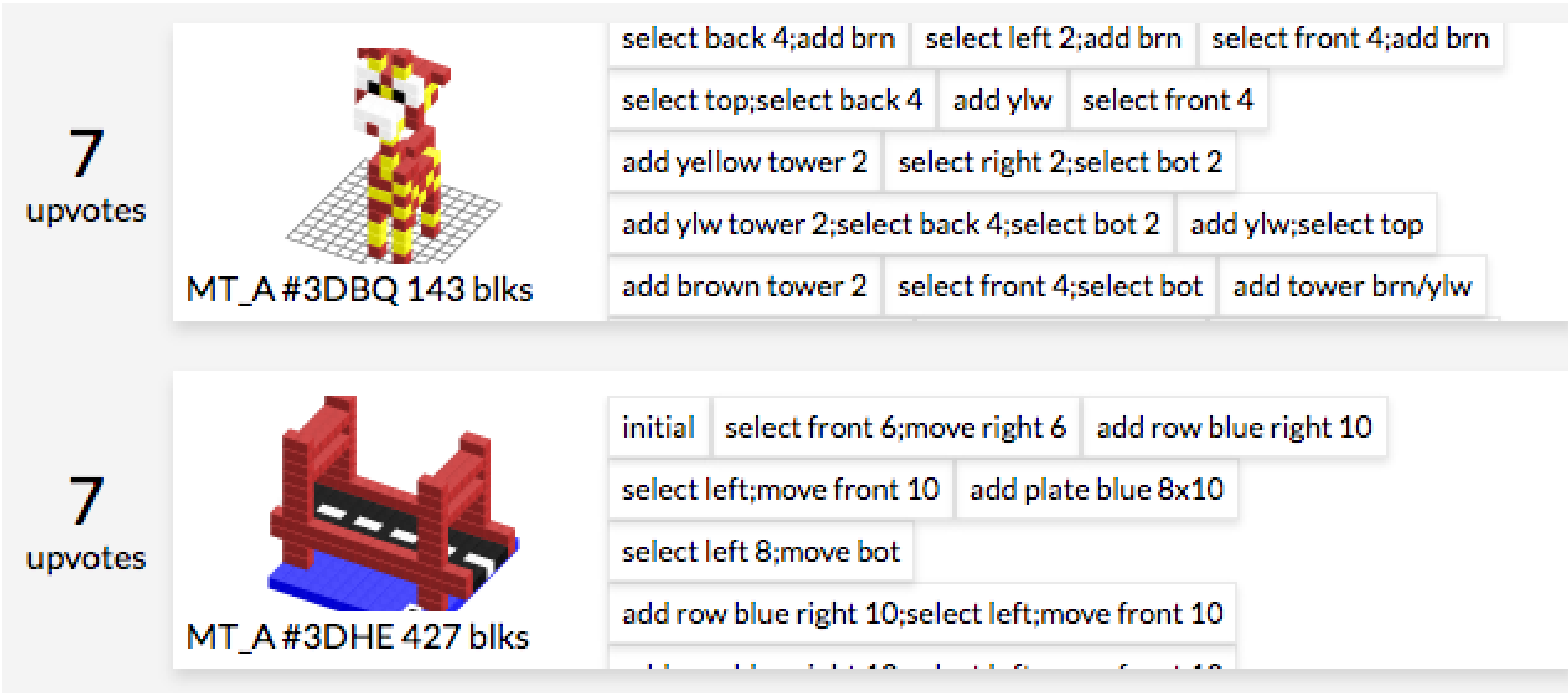
MT_A #3PWW 972 blks

initial | select left 6 | select front 8 | black 10x10x10 frame
black 10x10x10 frame | move front 10 | move left 9 | move bot 8
move front 7 | move left 9 | move front | move front 9
move left 9 | move bot 1 | black 10x10x10 frame

right 6; back 3; add yellow column 3

black 10x10x10 frame, green cube size 4

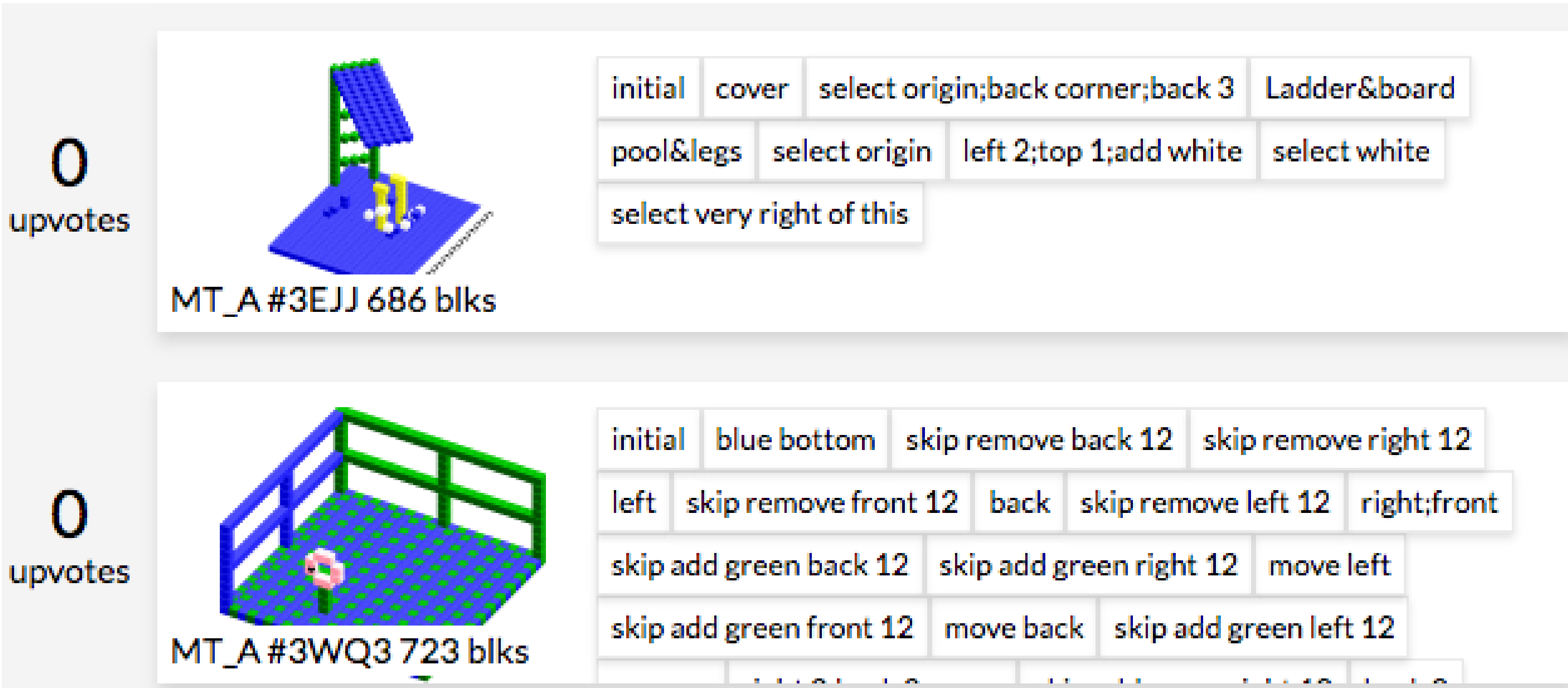
Leaderboard



add brn, add ylw, add brn tower 2

add plate 8x10

Leaderboard



skip add green back 12, skip remove back 12

Setup

- qualifier: build a fixed structure
- post-qual: over 3 days build whatever they want
- prizes for best structures
    - day 1: bridge, house, animal
    - day 2: tower, monster(s), flower(s)
    - day 3: ship(s), dancer(s), and castle
- prize for top h-index
    - a rule (and its author) gets a citation whenever it is used

Basic statistics

- 70 workers qualified, 42 participated, 230 structures

- 64075 utterances, 36589 accepts

    - each accept leads to a datapoint labeled by derivation(s)

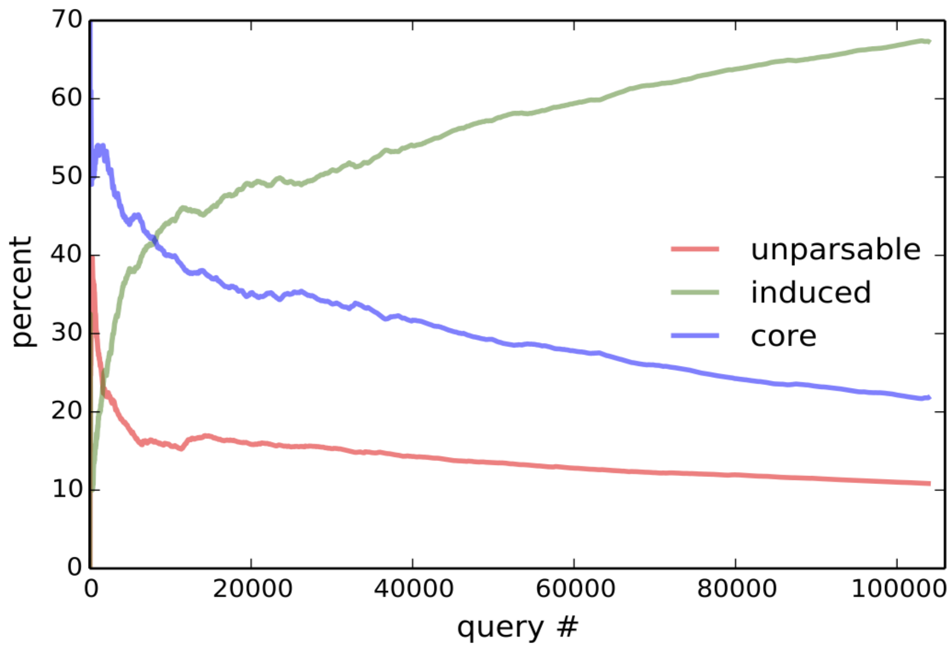
- 2495 definitions combining over 15k commands, 2817 induced rules (¡100 core)

Is naturalization happening

maybe best to use the core language and program...

Is naturalization happening

maybe best to use the core language and program...

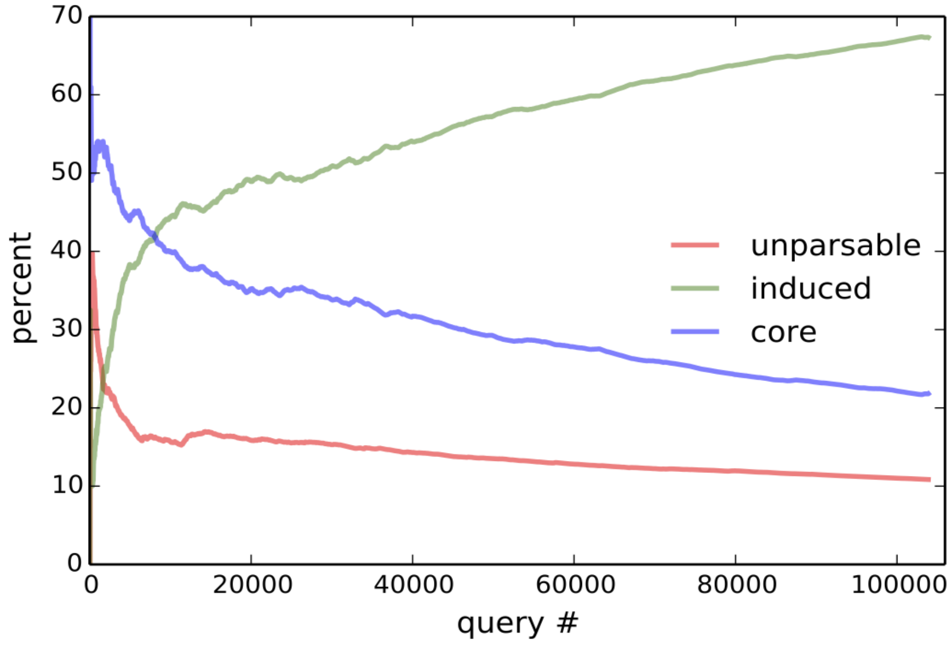


core: utterance parsable with the initial core grammar

induced: parsable with induced grammar by not by core

unparsable: not parsable at all

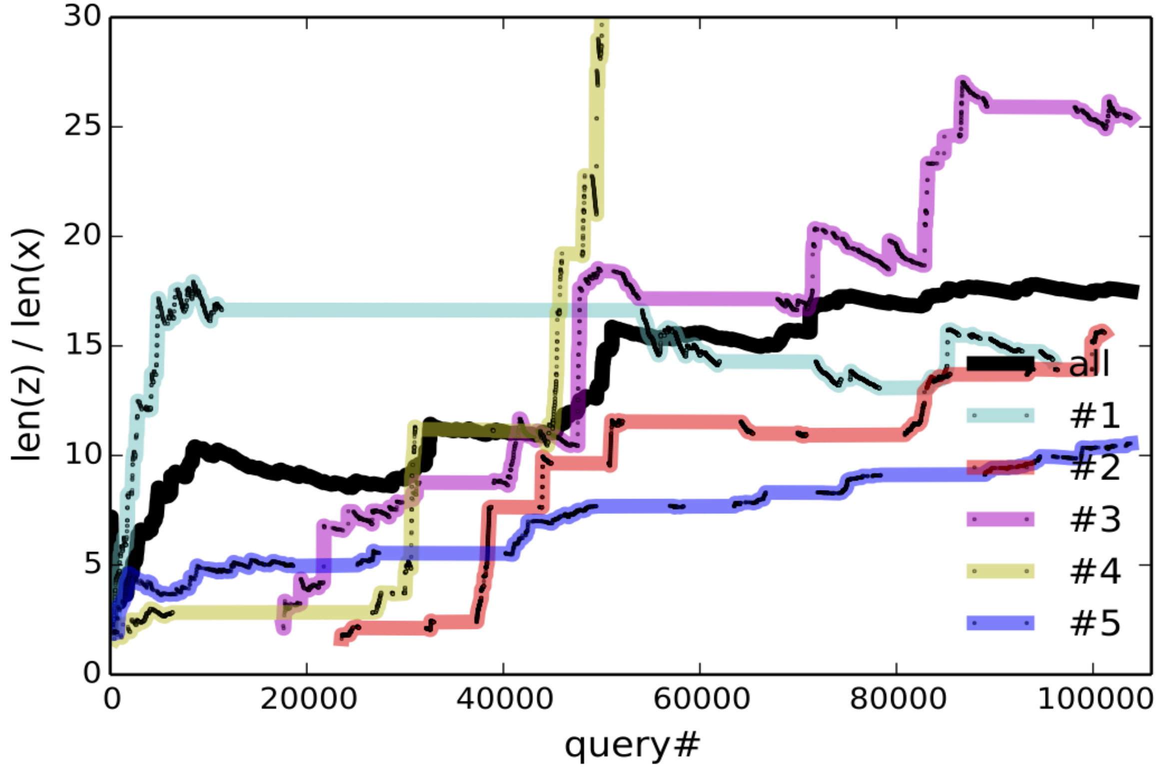
Is naturalization happening



naturalization:

- 67% of all at the end (up from 0 in the beginning)
- 72.9% of all accepted, and 85.9% of the last 10k accepted

Expressive power



- cumulative average of string.length in program / utterance
- len(z)/len(z) is very stable at 2 for core language
- varies greatly by user

Modes of naturalization

short forms:

left, l, mov left, go left, ¡, sel left

br, blk, blu, brn, orangeright, left3

add row brn left 5

:= *add row brown left 5*

Modes of naturalization

syntactic:

go down and right

$:=$ *go down; go right*

select orange

$:=$ *select has color orange*

add red top 4 times

$:=$ *repeat 4 [add red top]*

l white

$:=$ *go left and add white*

mov up 2

$:=$ *repeat 2 [select up]*

go up 3

$:=$ *go up 2; go up*

Modes of naturalization

higher level:

add black block width 2 length 2 height 3

:= {*repeat 3 [add black platform width 2...*

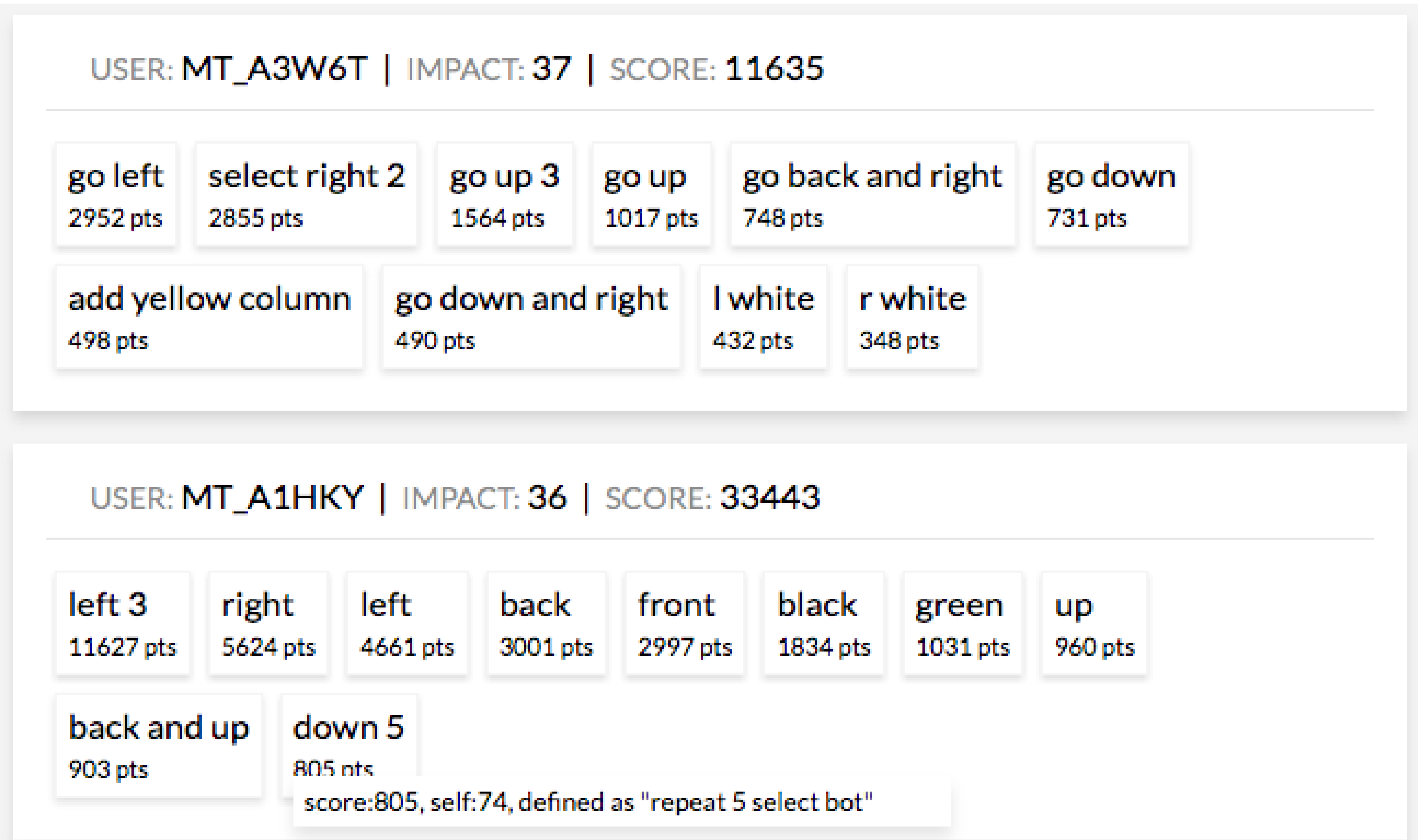
flower petals

:= *flower petal; back; flower petals*

red cube size 5, add green plate 2 x 4, 5 x 5 open green square, brownbase

Citations

an induced rule gets a citation whenever it is used in a structure



Citations

USER: MT_A37RS | IMPACT: 23 | SCORE: 3599

move left 6	780 pts
add brown tower 2	721 pts
mov up	577 pts
mov back	357 pts
move down 3	327 pts
mov right 3	291 pts
front 11	154 pts
add 6 brown front	138 pts
move up 2	129 pts
down	125 pts

USER: MT_A7HDY | IMPACT: 22 | SCORE: 2927

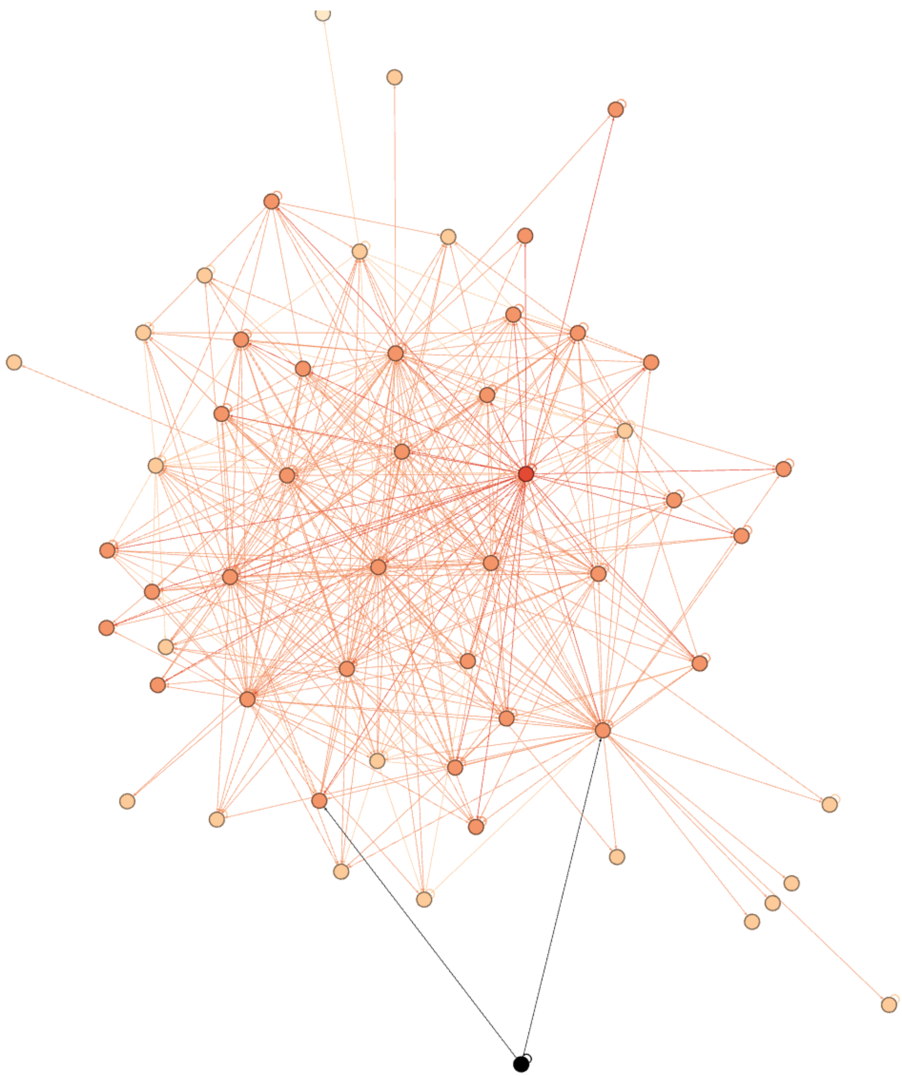
add b	1705 pts
add row brown right 9	236 pts
add tower brown 10	234 pts
add blu	172 pts
add blk	154 pts
add brn	144 pts
add row brn left 5	80 pts
add row brown left 3	75 pts
t	65 pts
l	62 pts

USER: MT_A25AX | IMPACT: 22 | SCORE: 1068

add brown col	243 pts
red line	149 pts
add grass	116 pts
add water	103 pts
5 x 5 open green square	88 pts
front 12	86 pts
short blue line	76 pts
open red 3 x 3 square	72 pts
add blue wall	68 pts
remove line	67 pts

Citations

basic statistics: 1113 cited rules, median 3, mean 46



left 3 : 5820 (*self*:459)

go up 3 : 712 (*self*:208)

right : 2879 (*self*:207)

add brown tower 2 : 721 (*self*:63)

r white : 175 (*self*:174)

add red top 4 times : 309

go back and right : 272

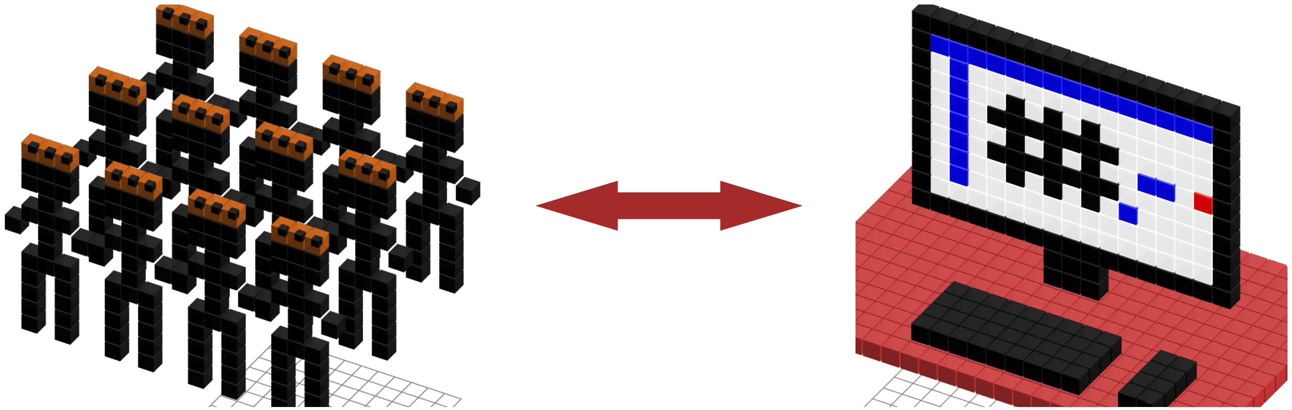
select orange : 256

add white plate 6 x 7 : 232

add brown row 3 : 203

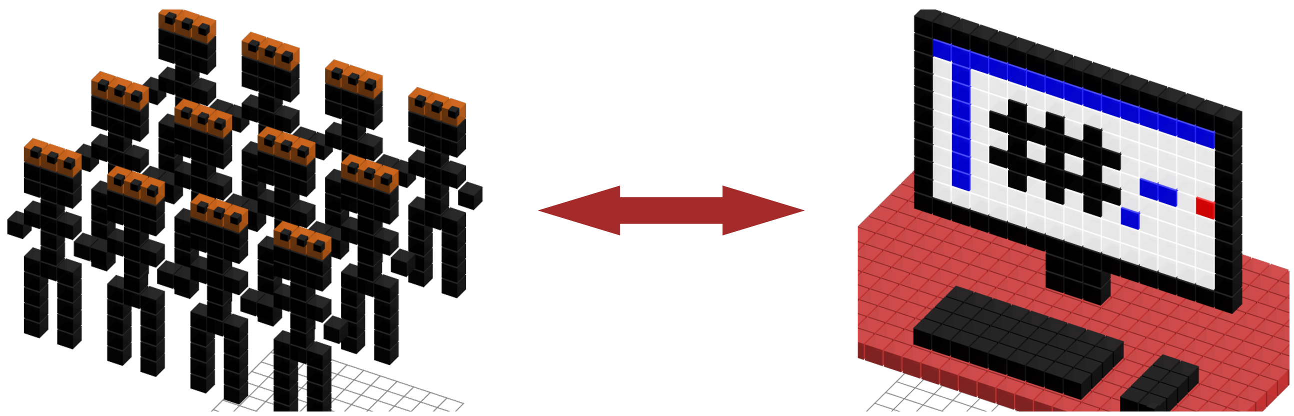
mov right 3 : 178

Naturalization via interactive learning



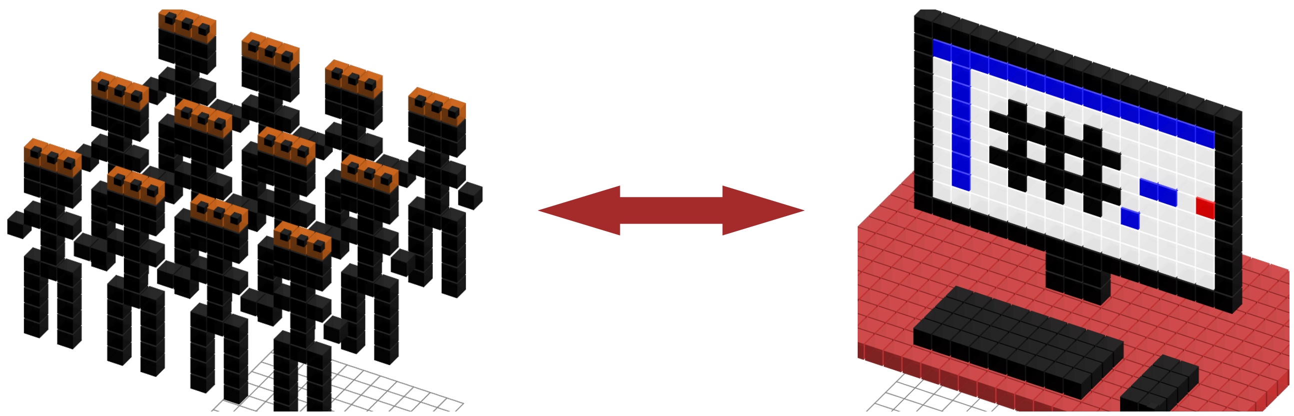
- community of users can build on each other to naturalize the core language

Naturalization via interactive learning



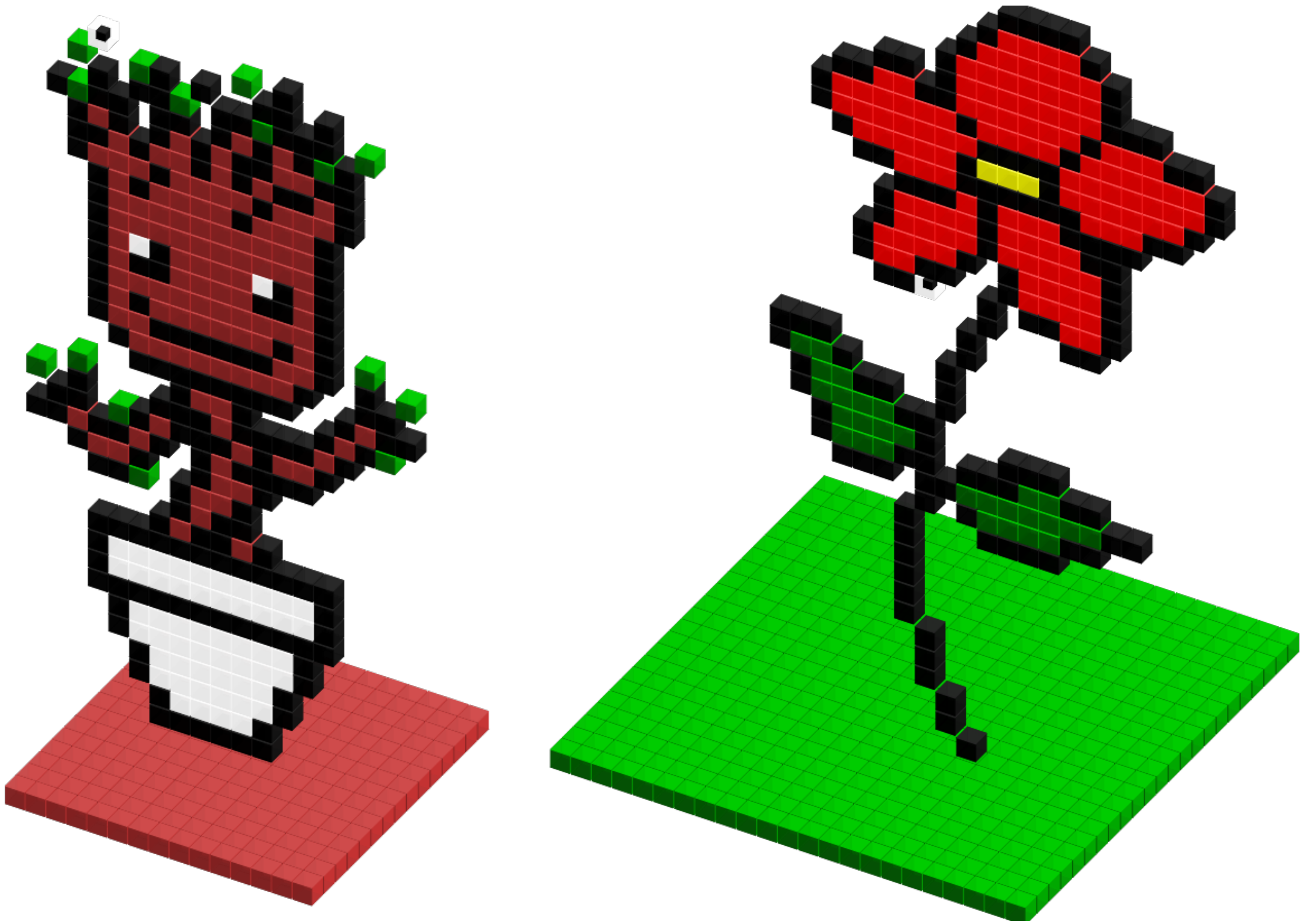
- community of users can build on each other to naturalize the core language
- crowdworkers are able to provide strong supervision using definitions

Naturalization via interactive learning

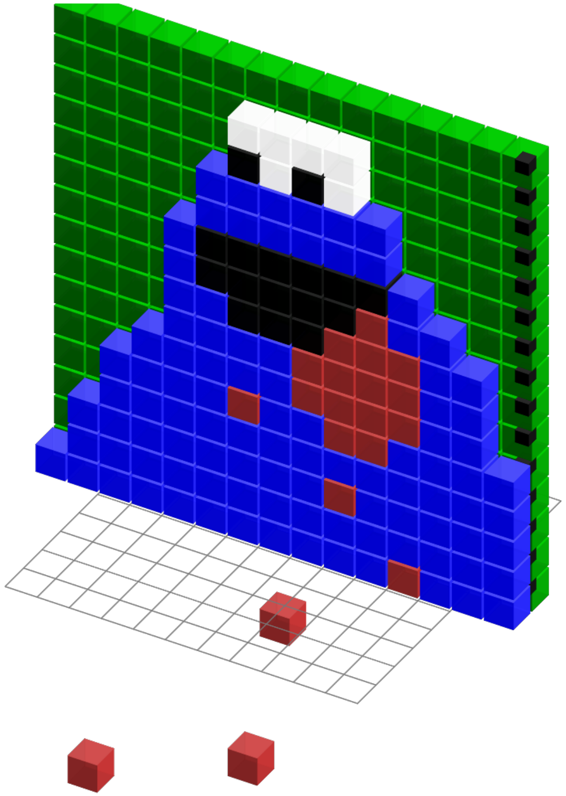


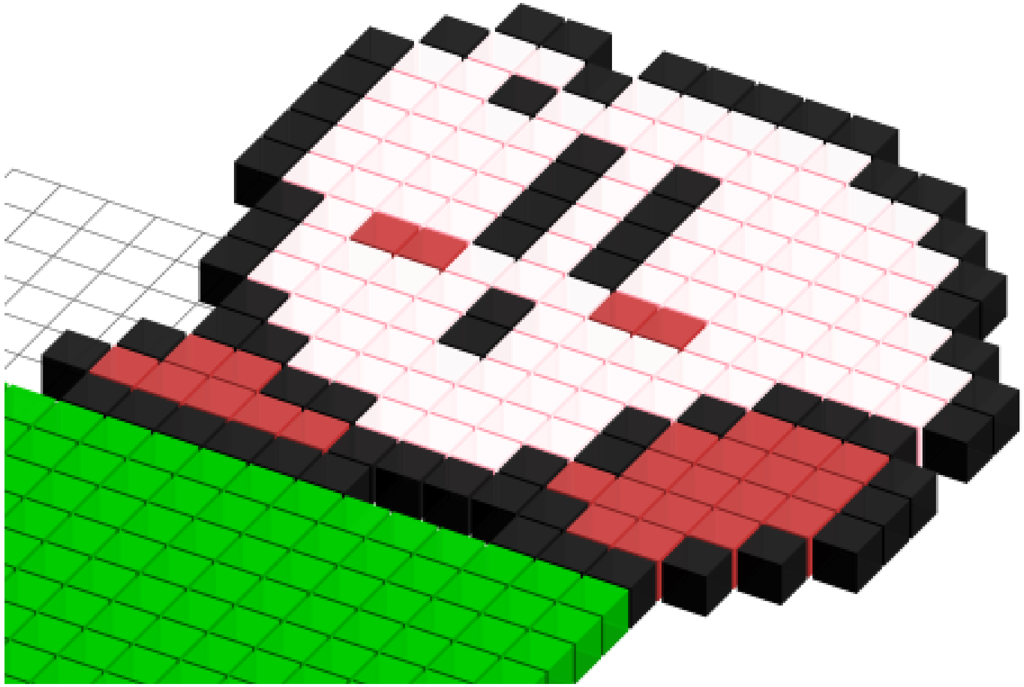
- community of users can build on each other to naturalize the core language
- crowdworkers are able to provide strong supervision using definitions
- from core language the users are forced to use, to a language users want to use

Thanks for listening



Questions?





Issues

- predictability and interpretability: 2k+ rules
- after 60k utterances, some simple utterances still not covered
- generalization of higher level concepts
  - chair $\rightarrow$ chair with red legs