Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions

Yoav Artzi and Luke Zettlemoyer
University of Washington

TACL VOL. 1(2013):49−62 / NAACL 2013

Semantic Parsing

Show me all papers about semantic parsing

$$\lambda x.paper(x) \land topic(x, SEMPAR)$$

Less Supervision

Answers
Demonstrations
Situated examples

More Domains

Databases
Large knowledge-bases
Instructions
Referring expressions
Regular expressions

Later this session

Semantic Parsing

Show me all papers about semantic parsing

$$\lambda x.paper(x) \wedge topic(x, SEMPAR)$$

Less Sup

More Domains

Modeling

Situated Parsing

Executing Navigation Instructions

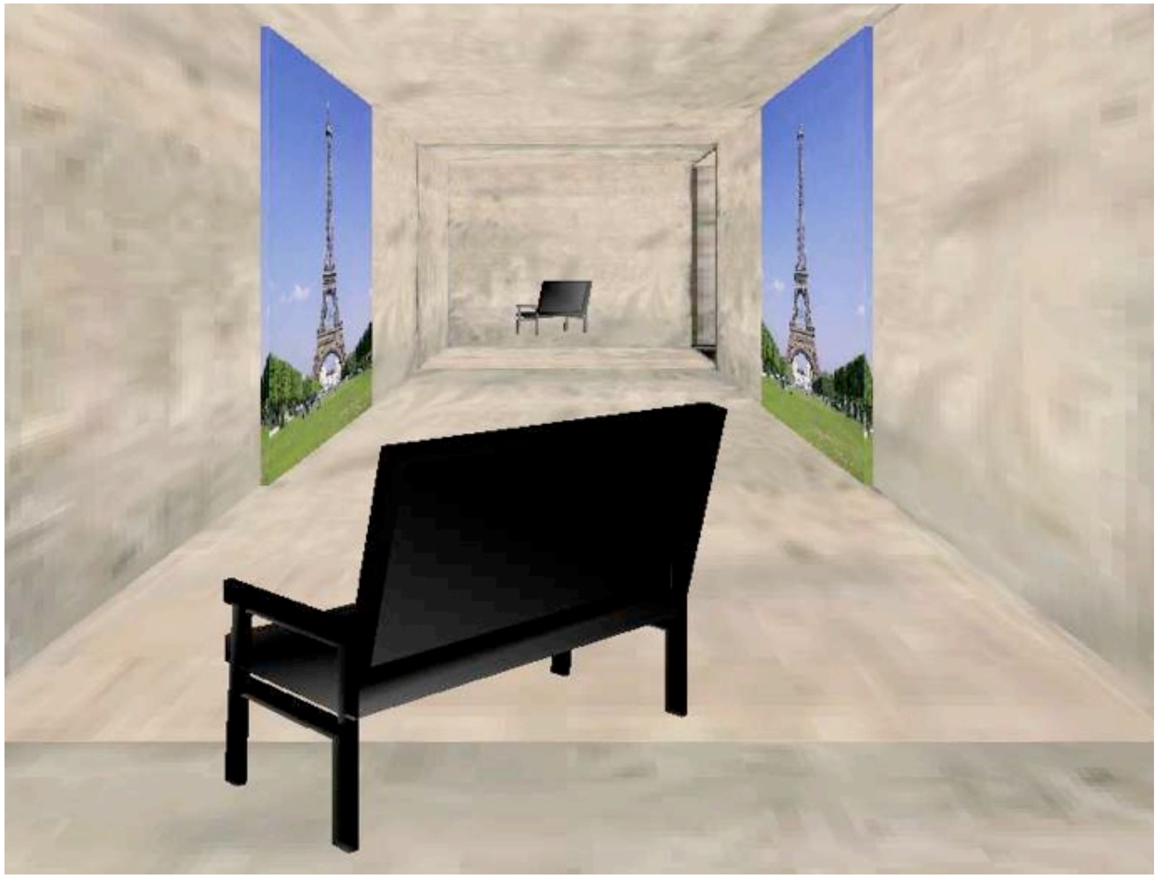
place your back against the wall of the t intersection

turn left

go forward along the pink flowered carpet hall two segments to the intersection with the brick hall

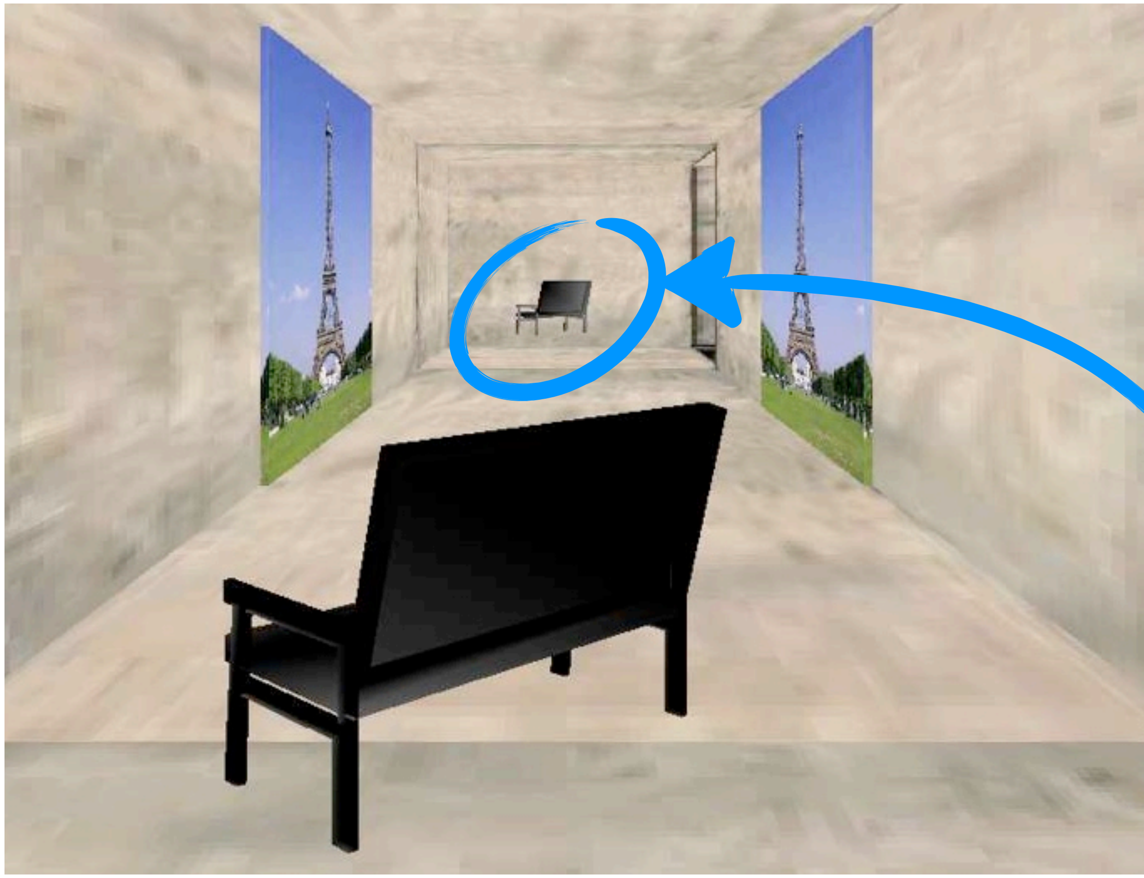
⋮

Resolve Referents



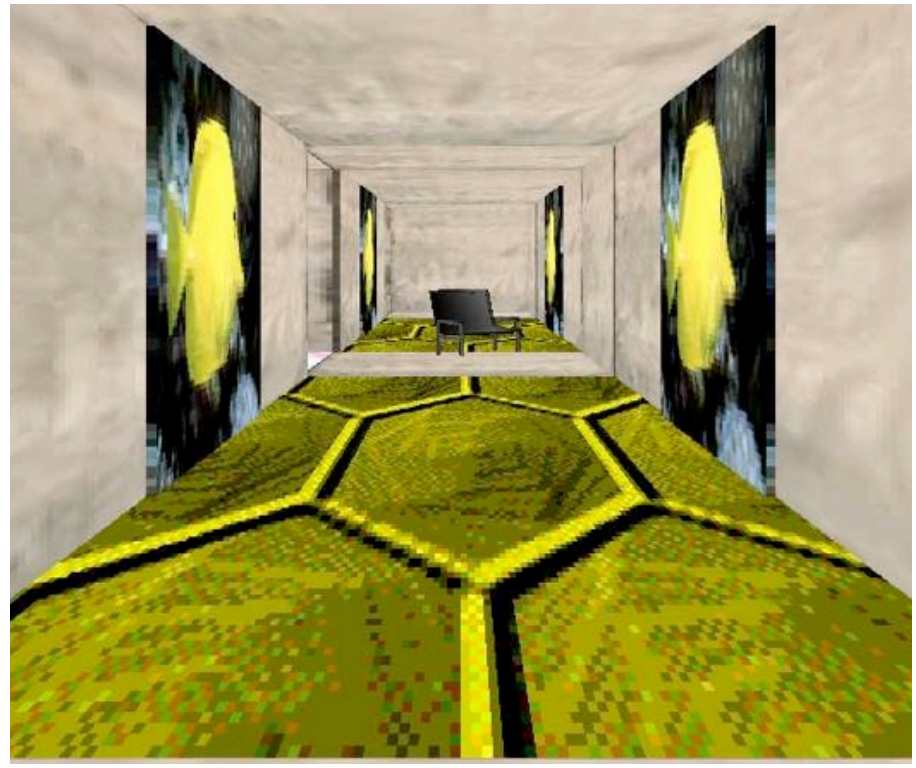
go to the next sofa

Resolve Referents



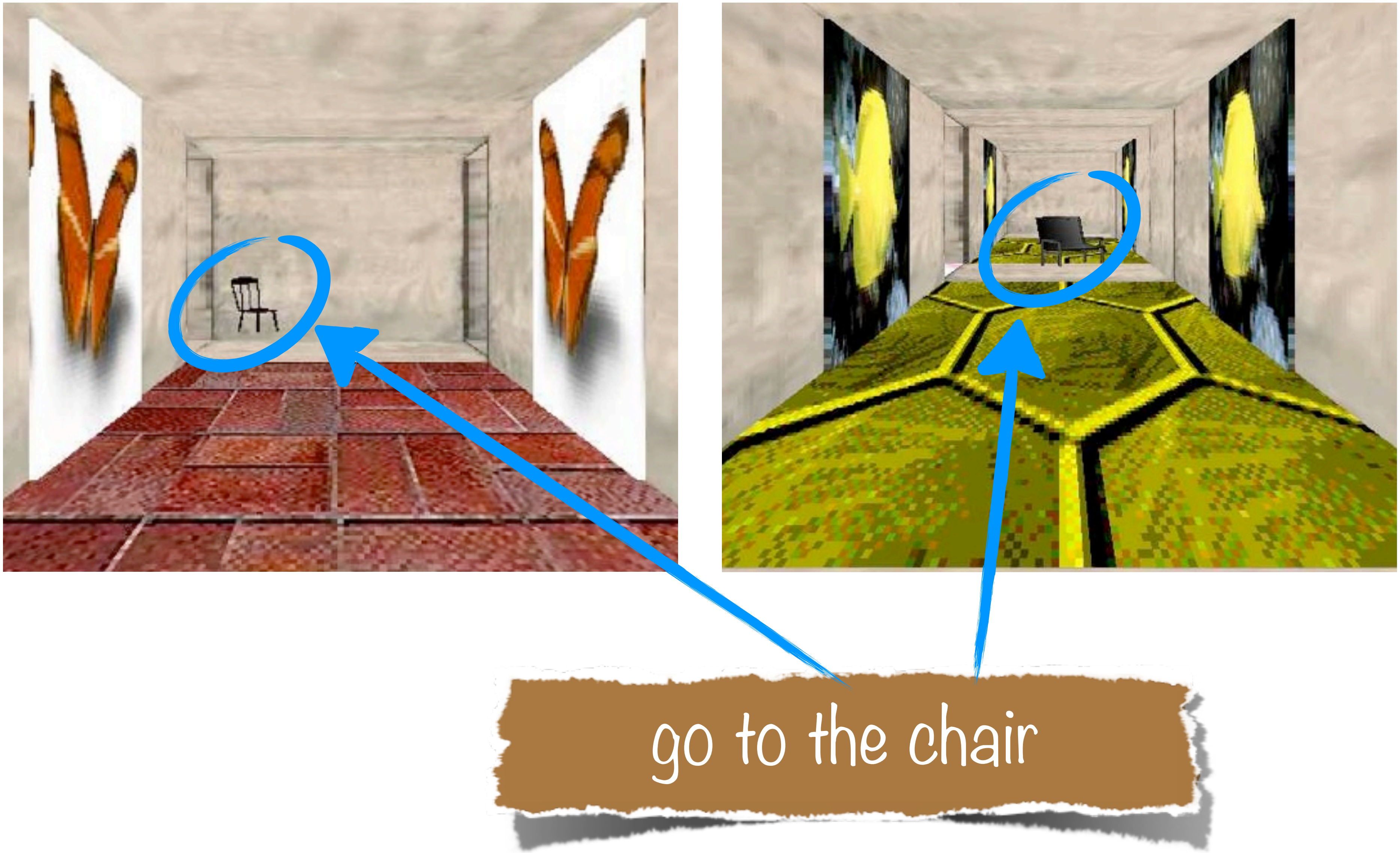
go to the next sofa

Disambiguate Word Sense

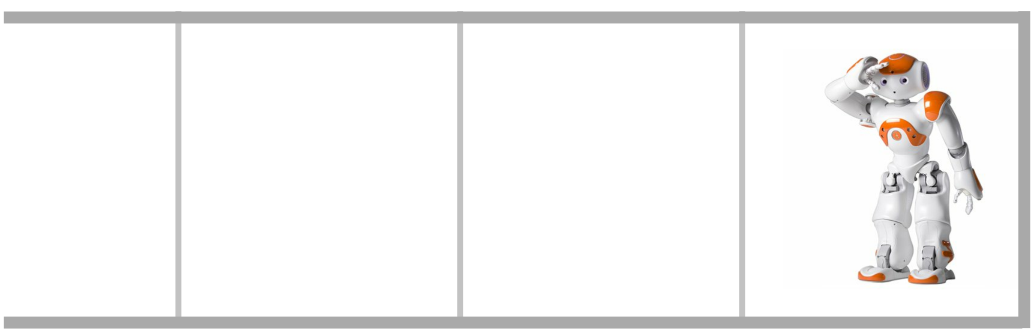


go to the chair

Disambiguate Word Sense

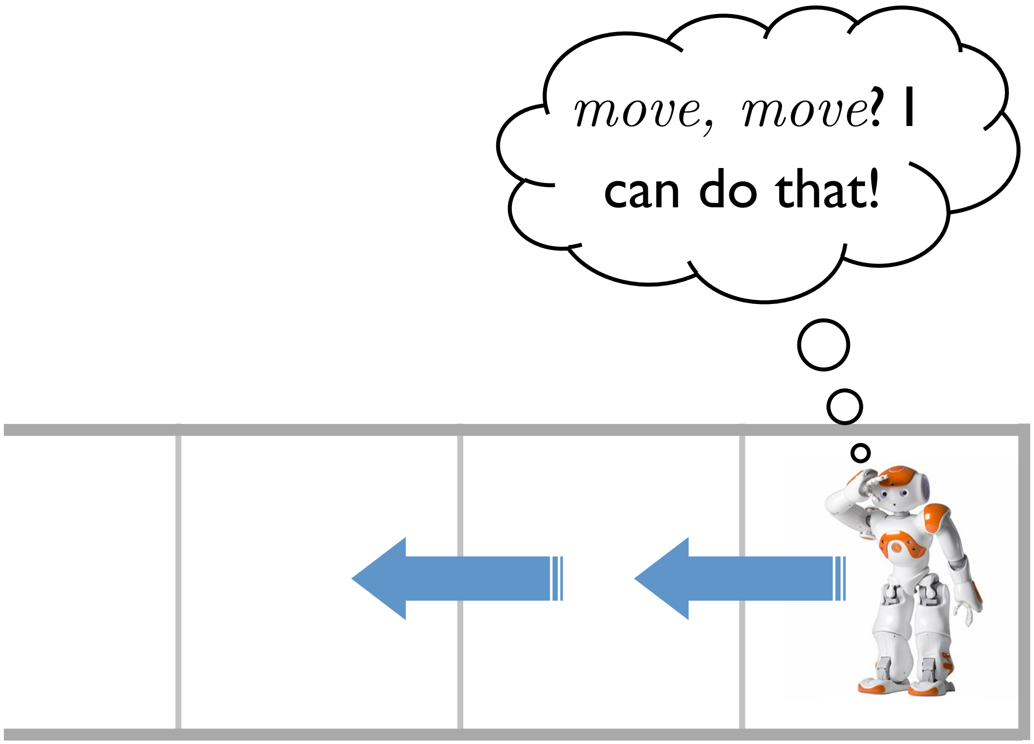


Identify Executable Actions



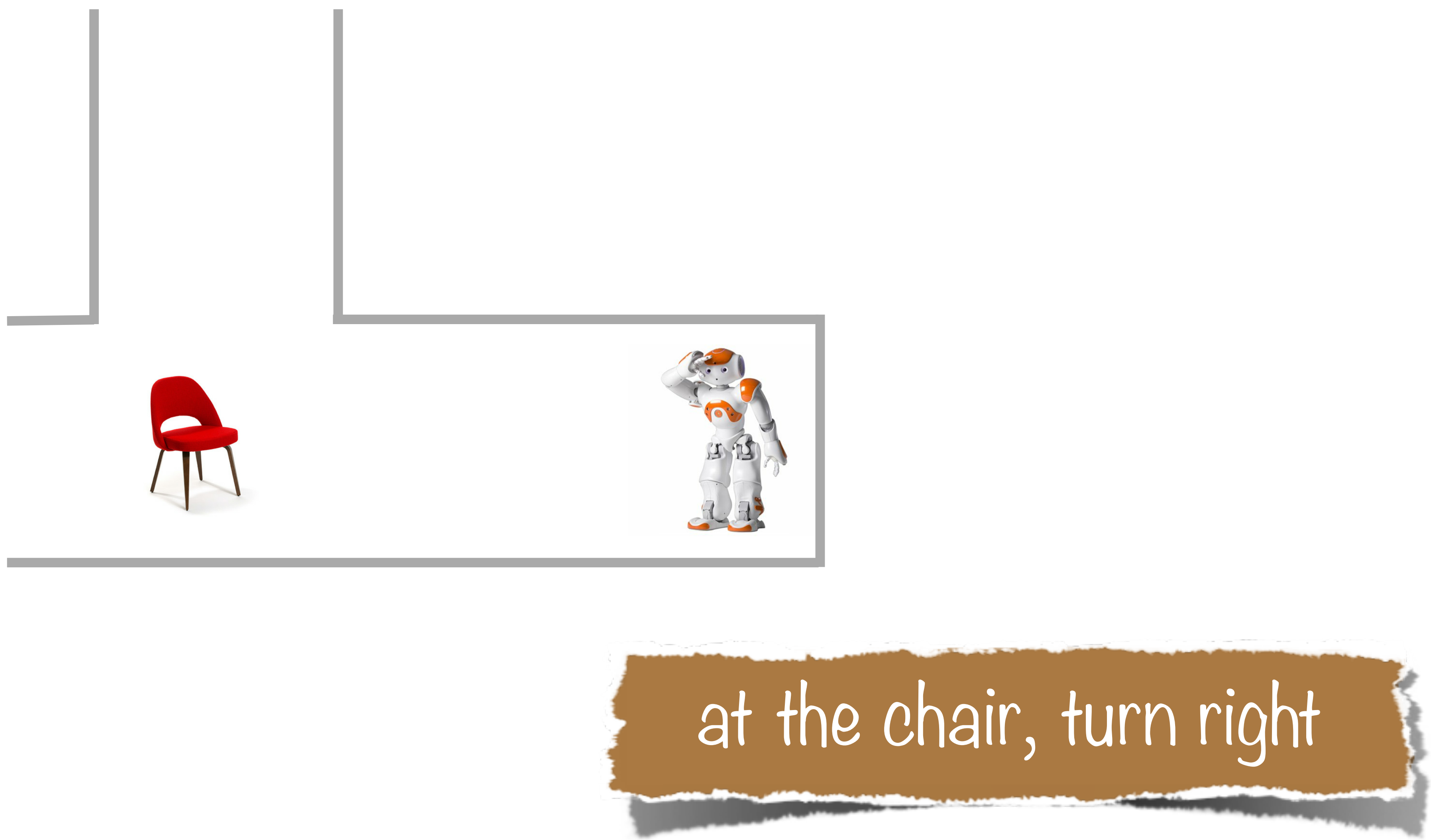
walk forward twice

Identify Executable Actions

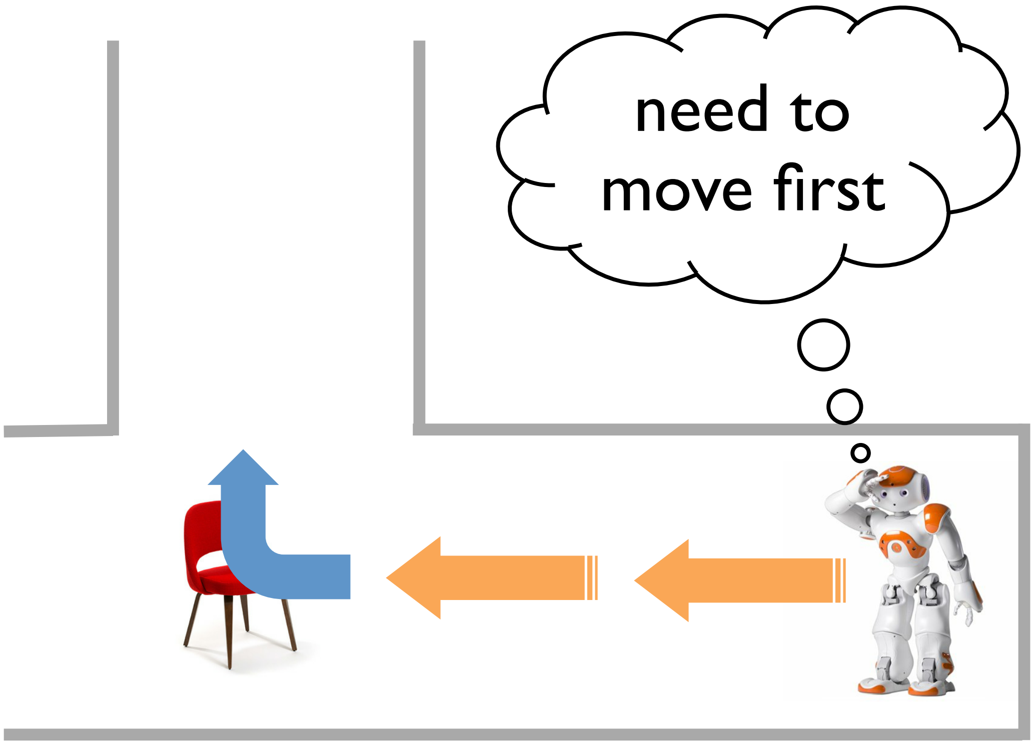


walk forward twice

Understand Implicit Requests

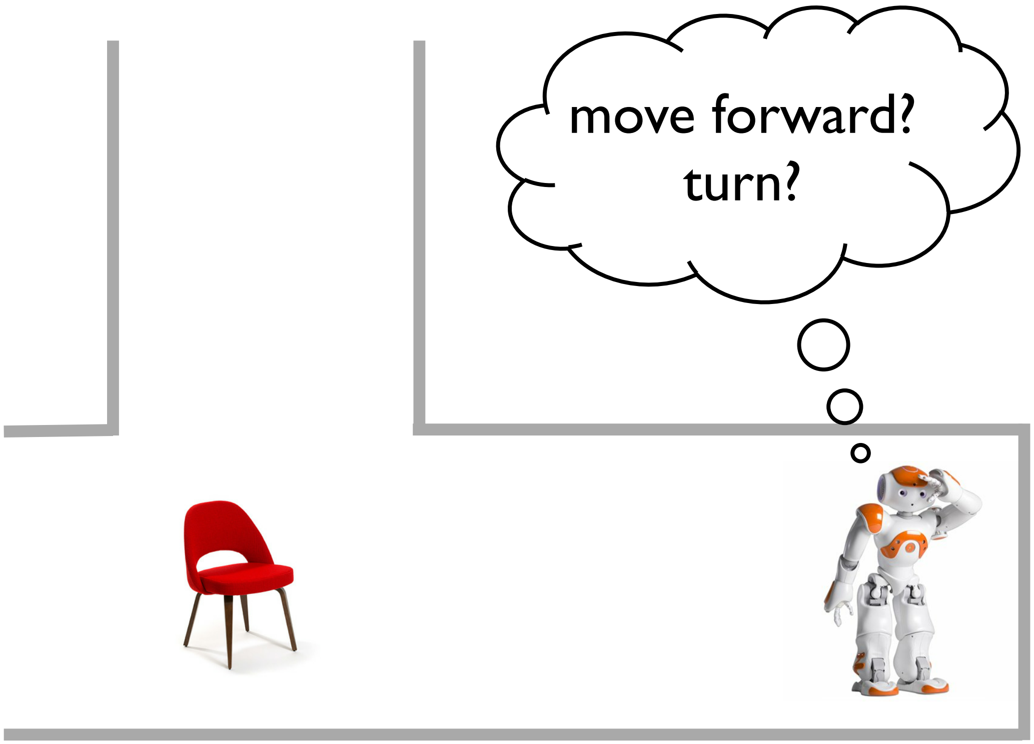


Understand Implicit Requests



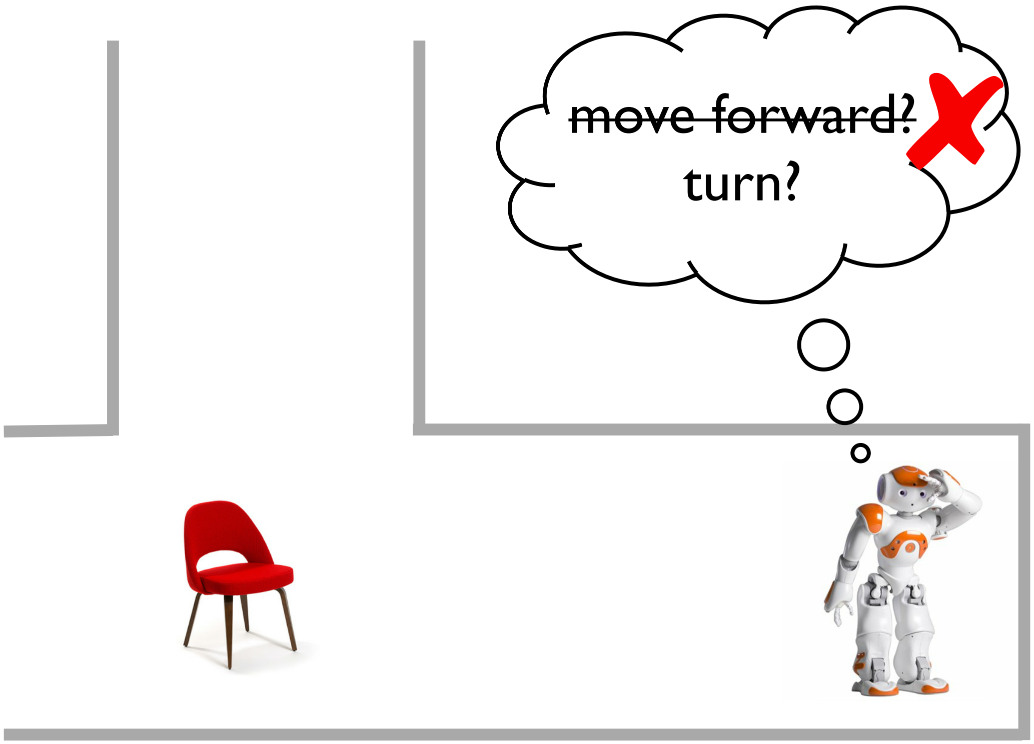
at the chair, turn right

Grounded Learning



turn to face the chair

Grounded Learning



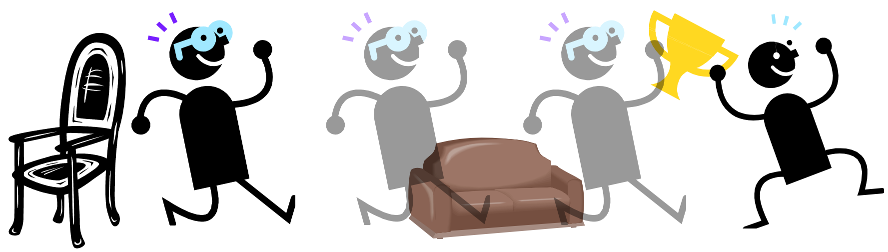
turn to face the chair

Learning Signal

Instruction:

at the chair, move forward three steps past the sofa

Demonstration:

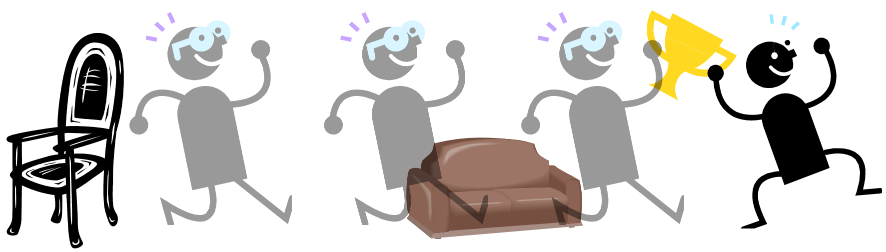


Learning Signal

Instruction:

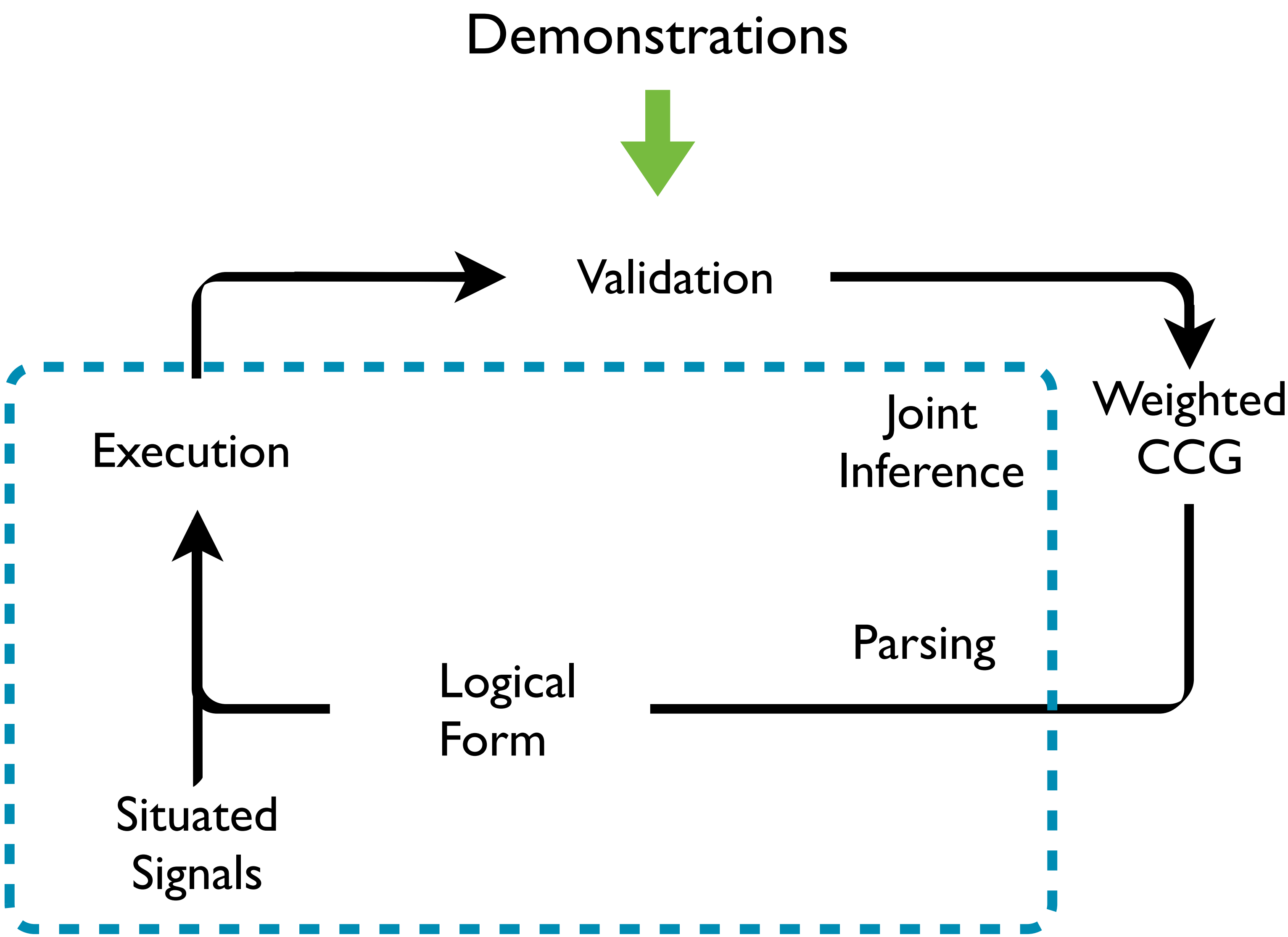
at the chair, move forward three steps past the sofa

Demonstration:

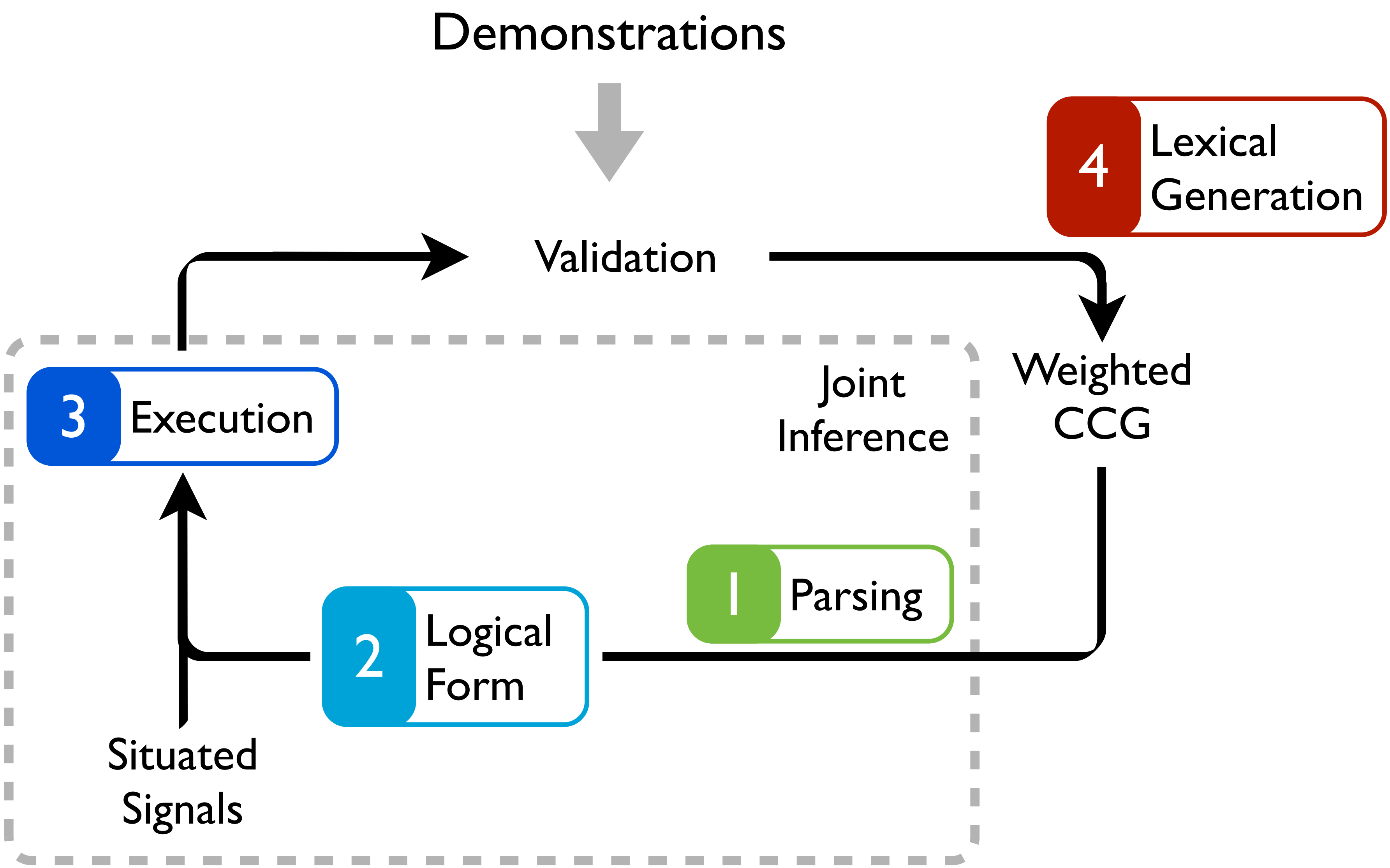


<u>During learning</u>: validate executions of different interpretations against demonstrations

Learning


Demonstrations
Validation
Weighted
CCG
Joint
Inference
Execution
Parsing
Logical
Form
Situated
Signals

Learning


Demonstrations
Validation
4 Lexical Generation
Weighted CCG
3 Execution
Joint Inference
1 Parsing
2 Logical Form
Situated Signals

Combinatory Categorial Grammars

$$
\begin{array}{cccc}
\text{move} & \text{to} & \text{the} & \text{chair} \\
\hline
S & AP/NP & NP/N & N \\
\lambda a.move(a) & \lambda x.\lambda a.to(a,x) & \lambda f.\iota x.f(x) & \lambda x.chair(x)
\end{array}
$$

$$
\frac{NP/N \quad N}{NP} > \qquad \iota x.chair(x)
$$

$$
\frac{AP/NP \quad NP}{AP} > \qquad \lambda a.to(a, \iota x.chair(x))
$$

$$
\frac{AP}{S\backslash S} \qquad \lambda f.\lambda a.f(a) \wedge to(a, \iota x.chair(x))
$$

$$
\frac{S \quad S\backslash S}{S} > \qquad \lambda a.move(a) \wedge to(a, \iota x.chair(x))
$$

Combinatory Categorial Grammars

$$
\begin{array}{cccc}
\text{move} & \text{to} & \text{the} & \text{chair} \\
\hline
S & AP/NP & NP/N & N \\
\lambda a.move(a) & \lambda x.\lambda a.to(a,x) & \lambda f.\iota x.f(x) & \lambda x.chair(x)
\end{array}
$$

$$\xrightarrow{\quad\quad\quad\quad\quad\quad\quad\quad} >$$

$$NP$$
$$\iota x.chair(x)$$

$$\xrightarrow{\quad\quad\quad\quad\quad\quad\quad\quad} >$$

$$AP$$
$$\lambda a.to(a, \iota x.chair(x))$$

$$\overline{\quad\quad\quad\quad\quad\quad\quad\quad}$$

$$S\backslash S$$
$$\lambda f.\lambda a.f(a) \wedge to(a, \iota x.chair(x))$$

$$\xrightarrow{\quad\quad\quad\quad\quad\quad\quad\quad} >$$

$$S$$
$$\lambda a.move(a) \wedge to(a, \iota x.chair(x))$$

Lexicon | **Combinators**

Combinatory Categorial Grammars

x: move | to | the | cl

y:

$$\frac{\text{move}}{\begin{array}{c} S \\ \lambda a.move(a) \end{array}} \quad \frac{\text{to}}{\begin{array}{c} AP/NP \\ \lambda x.\lambda a.to(a,x) \end{array}} \quad \frac{\text{the}}{\begin{array}{c} NP/N \\ \lambda f.\iota x.f(x) \end{array}} \quad \frac{\text{cl}}{\lambda x.cl}$$

$$\frac{}{\begin{array}{c} NP \\ \iota x.chair(x) \end{array}}$$

$$\frac{}{\begin{array}{c} AP \\ \lambda a.to(a, \iota x.chair(x)) \end{array}}$$

$$\frac{}{\begin{array}{c} S \backslash S \\ \lambda f.\lambda a.f(a) \wedge to(a, \iota x.chair(x \end{array}}$$

$$\frac{}{\begin{array}{c} S \\ \lambda a.move(a) \wedge to(a, \iota x.chair(x)) \end{array}}$$

z: $\lambda a.move(a) \wedge to(a, \iota x.chair(x))$

Lexicon

Combinators

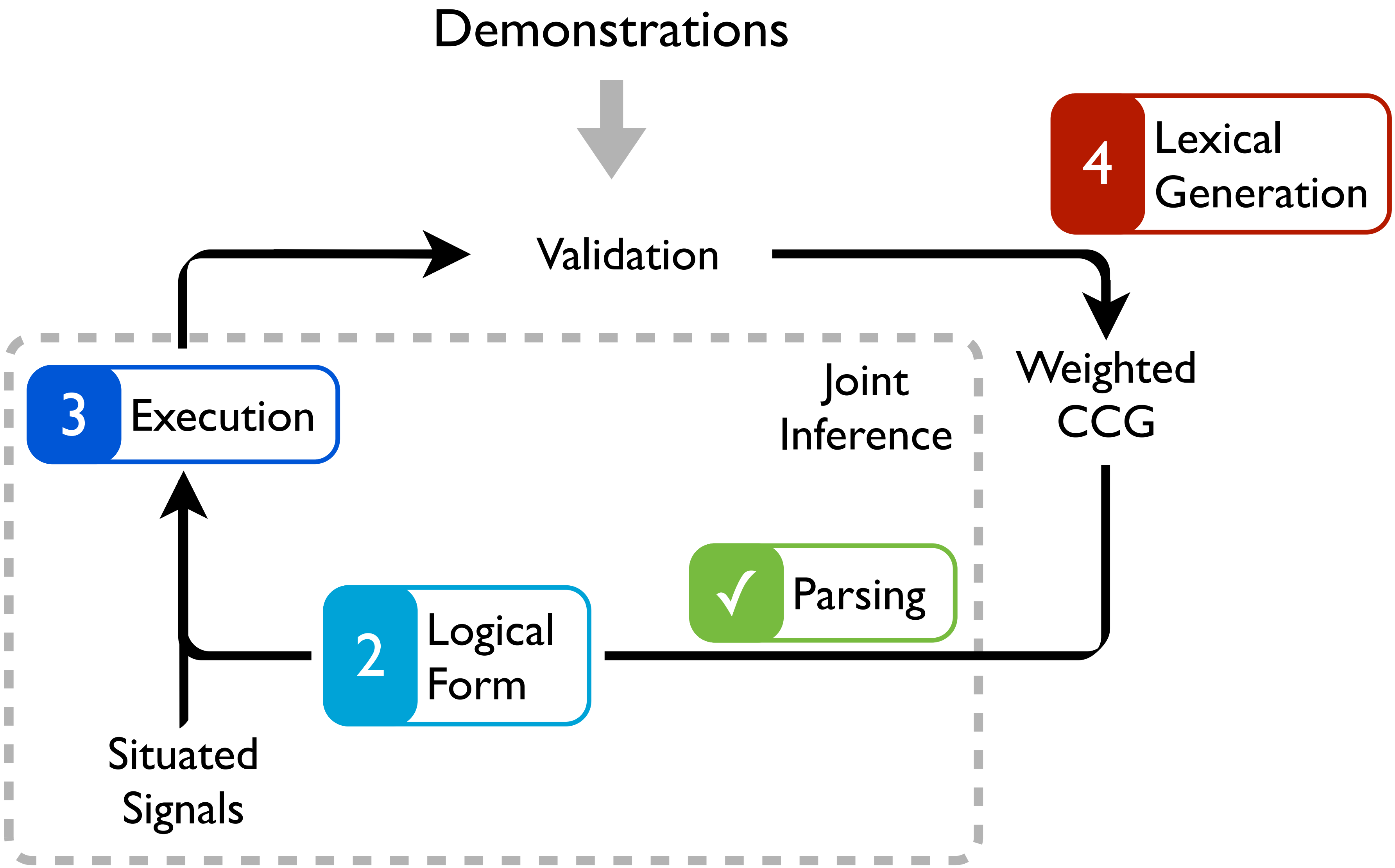
Weighted Linear CCGs

- Given a weighted linear model:
  - CCG lexicon Λ
  - Feature function $f : X \times Y \to \mathbb{R}^m$
  - Weights $w \in \mathbb{R}^m$
- The best parse is:

$$y^* = \arg\max_y w \cdot f(x, y)$$

- We consider all possible parses y for sentence x given the lexicon Λ

Learning



Demonstrations
Validation
4
Lexical Generation
Weighted CCG
3
Execution
Joint Inference
Parsing
2
Logical Form
Situated Signals

Modeling for Semantic Parsing

Nouns	Sets of entities
PPs and adjectives	Constrain sets
Noun phrases	Specific entities
Verbs	Relations between entities

Modeling for Semantic Parsing

Nouns	Sets of entities
PPs and adjectives	Constrain sets
Noun phrases	Specific entities
Verbs	Relations between entities

Works well for natural language interfaces for DBs

Modeling for Semantic Parsing

Nouns	Sets of entities
PPs and adjectives	Constrain sets
Noun phrases	Specific entities
Verbs	Relations between entities

Previous work on instructional language adopted <u>procedural representation</u>

[Matuszek et al. 2010; 2012; Chen, Mooney 2011; Chen 2012; Kim, Mooney 2012]

Modeling for Semantic Parsing

Nouns	Sets of entities
PPs and adjectives	Constrain sets
Noun phrases	Specific entities
Verbs	Relations between entities

Previous work on instructional language adopted <u>procedural representation</u>

How can we use this approach for instructions?

Modeling for Semantic Parsing

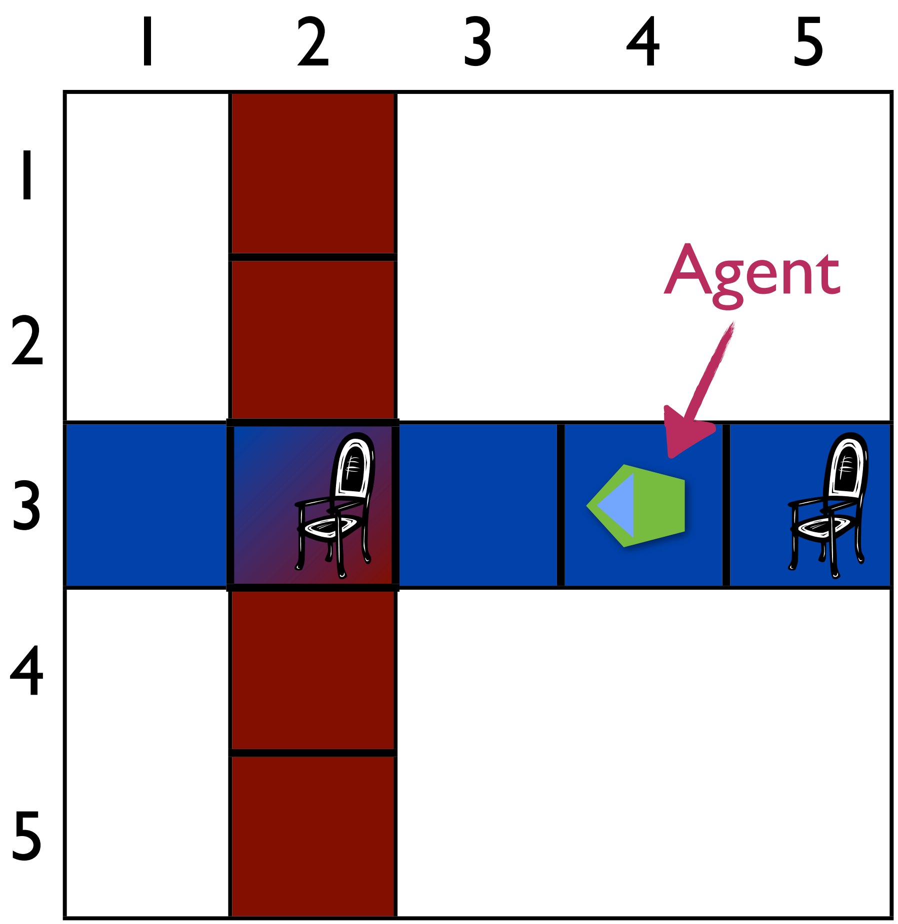
Name objects

Nouns	Sets of entities
PPs and adjectives	Constrain sets
Noun phrases	Specific entities

Instructions to execute

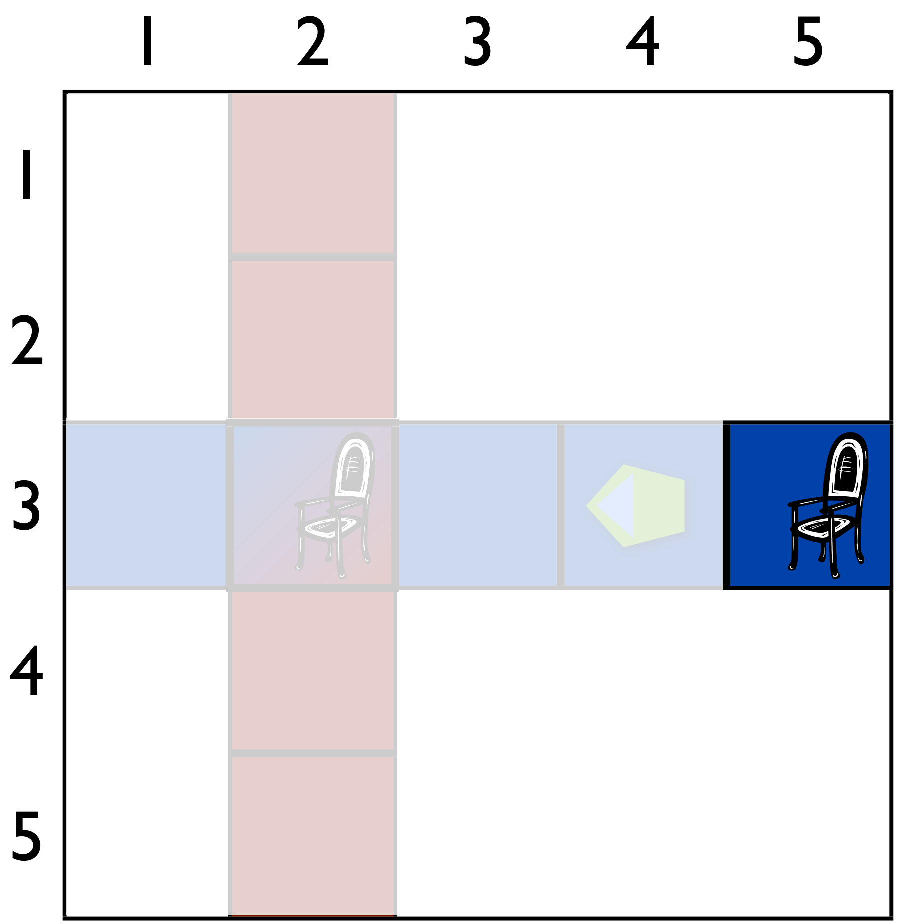
Verbs	Davidsonian Events
Imperatives	Sets of events

Spatial Environment Modeling



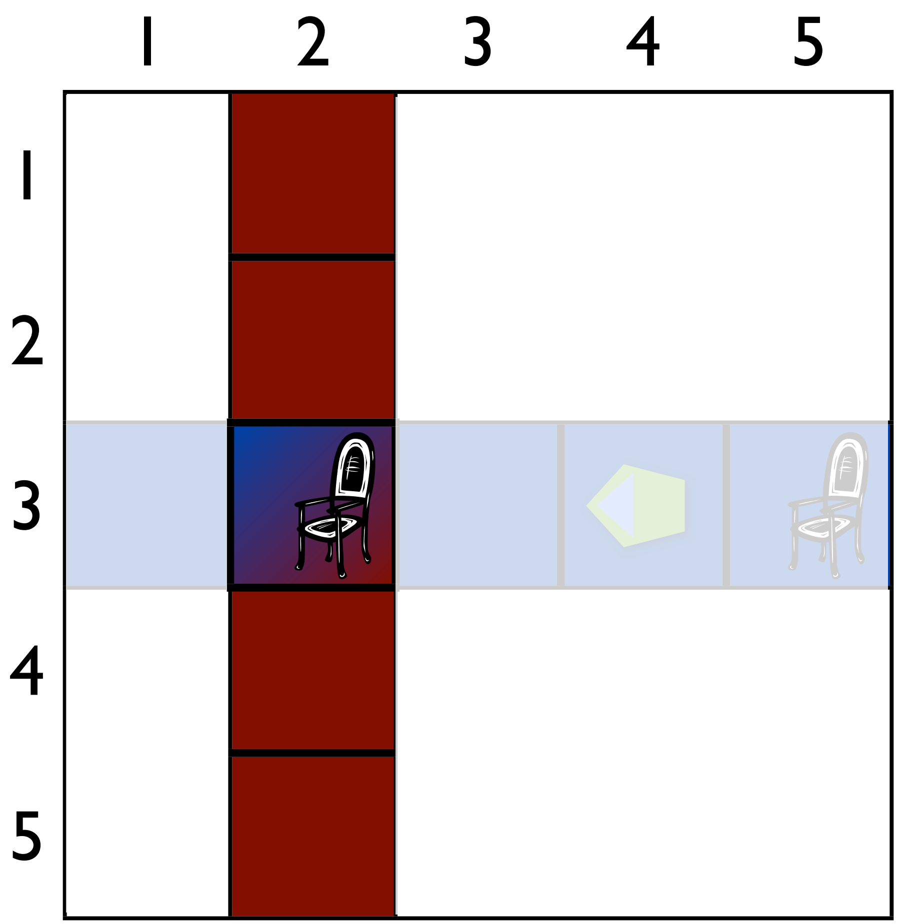
- Maps are graphs of connected positions
- Agent can move forward, turn right and turn left
- Agent perceives clusters of positions
- Clusters capture objects

Spatial Environment Modeling



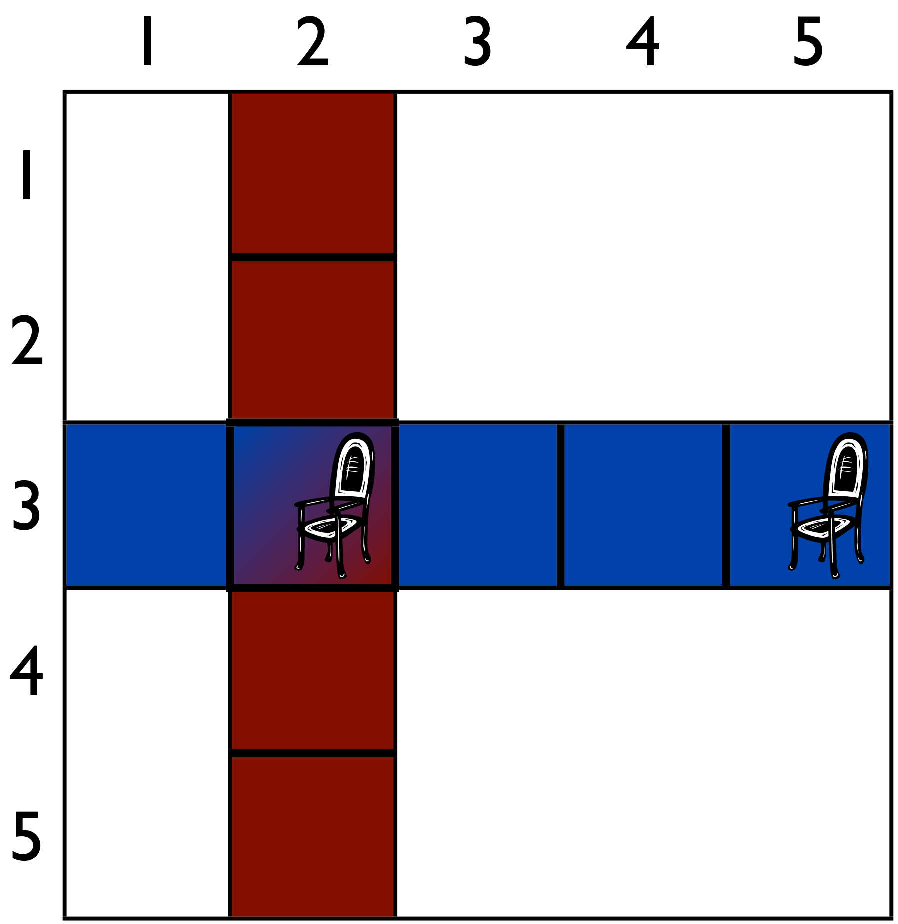
- Maps are graphs of connected positions
- Agent can move forward, turn right and turn left
- Agent perceives clusters of positions
- Clusters capture objects

Spatial Environment Modeling



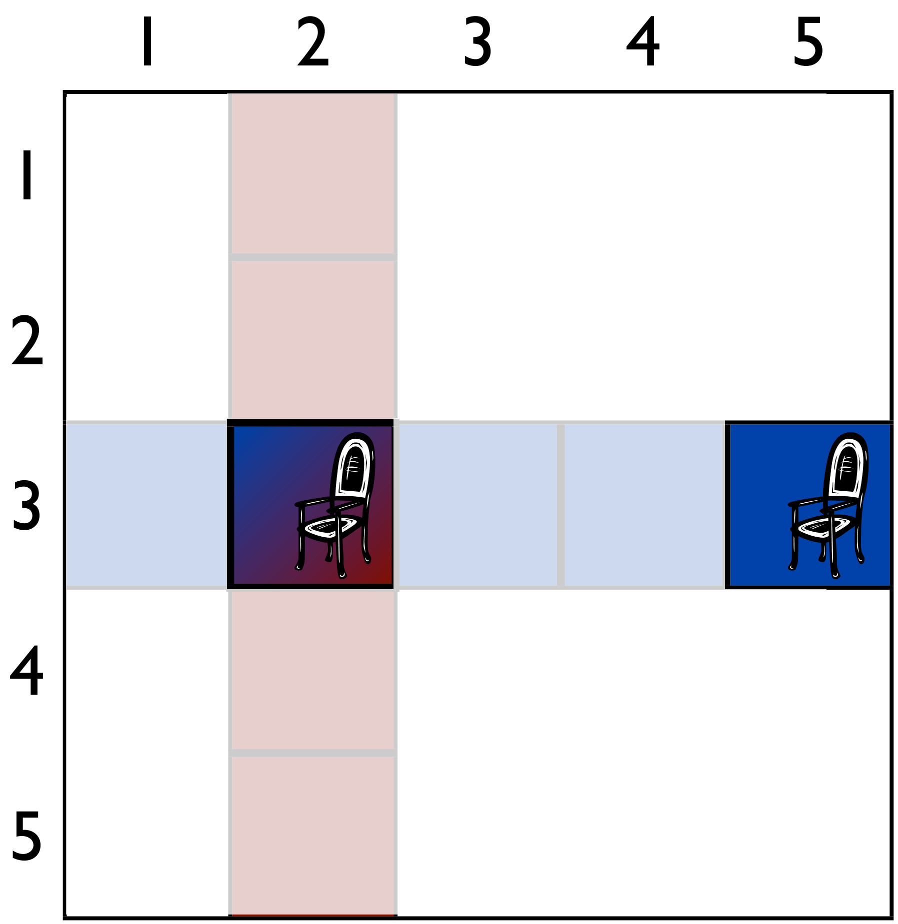
- Maps are graphs of connected positions
- Agent can move forward, turn right and turn left
- Agent perceives clusters of positions
- Clusters capture objects

Spatial Language



- Nouns
- Noun phrases
- Adjectives
- Prepositional phrases
- Spatial relations

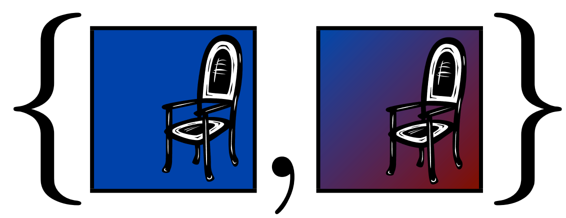
Spatial Language



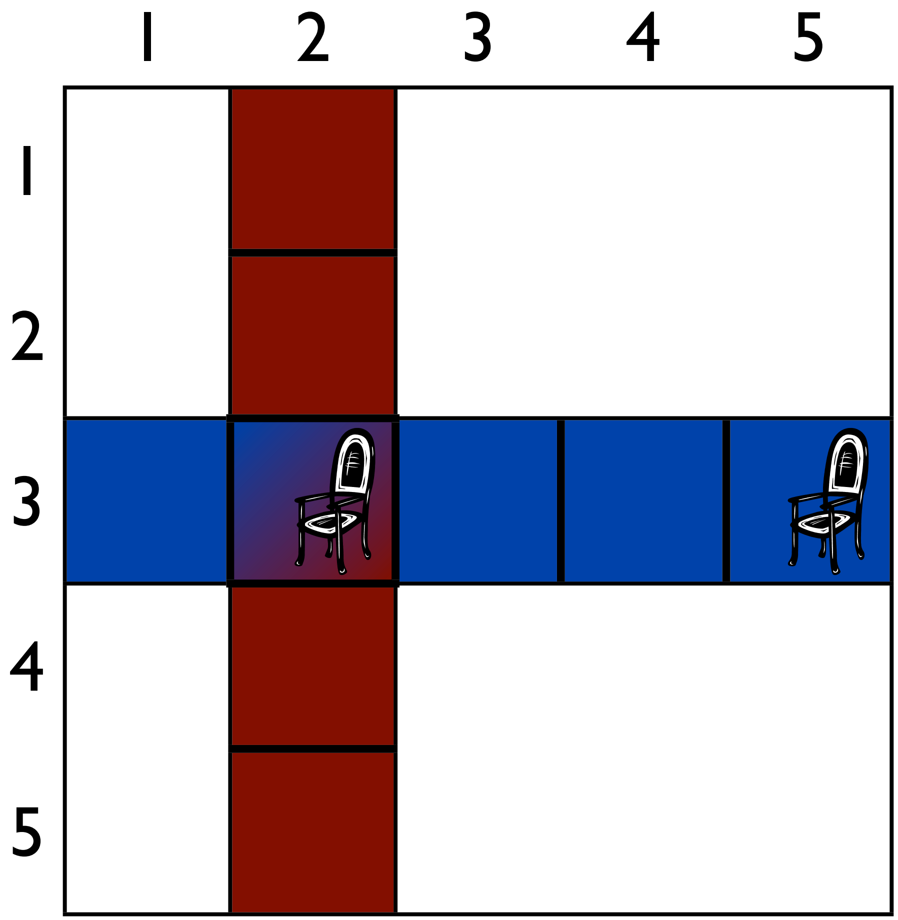
Nouns denote sets of objects

chair

$\lambda x.chair(x)$

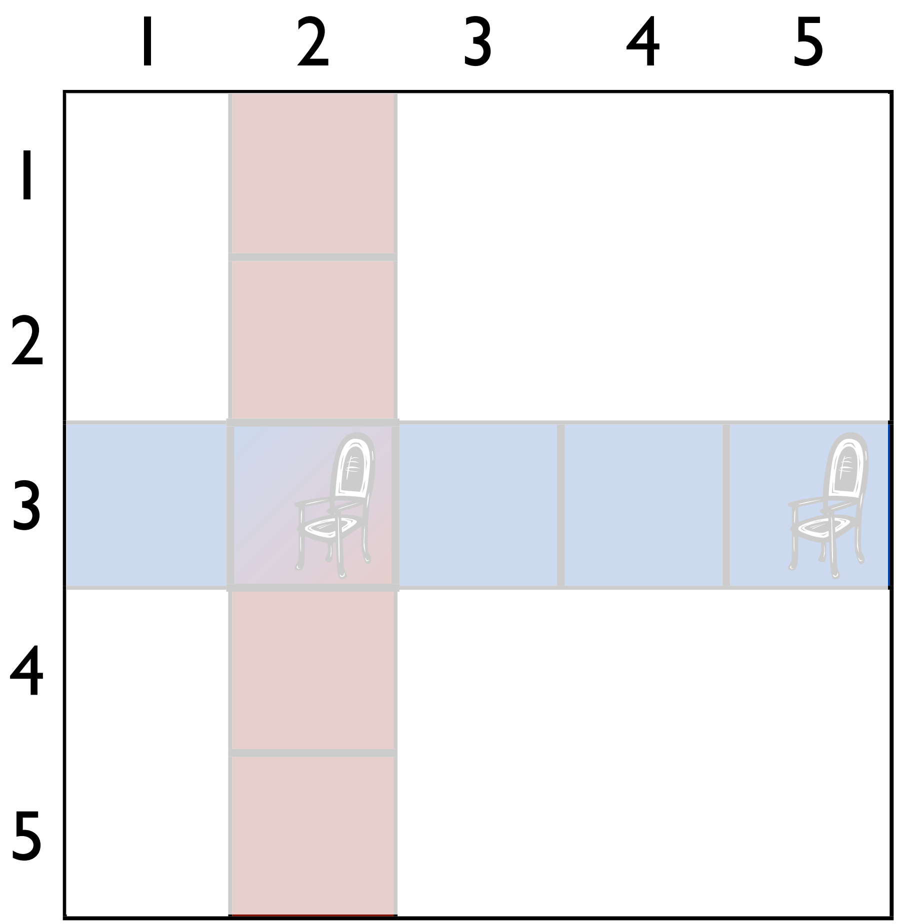


Spatial Language



Noun phrases name specific entities

Spatial Language

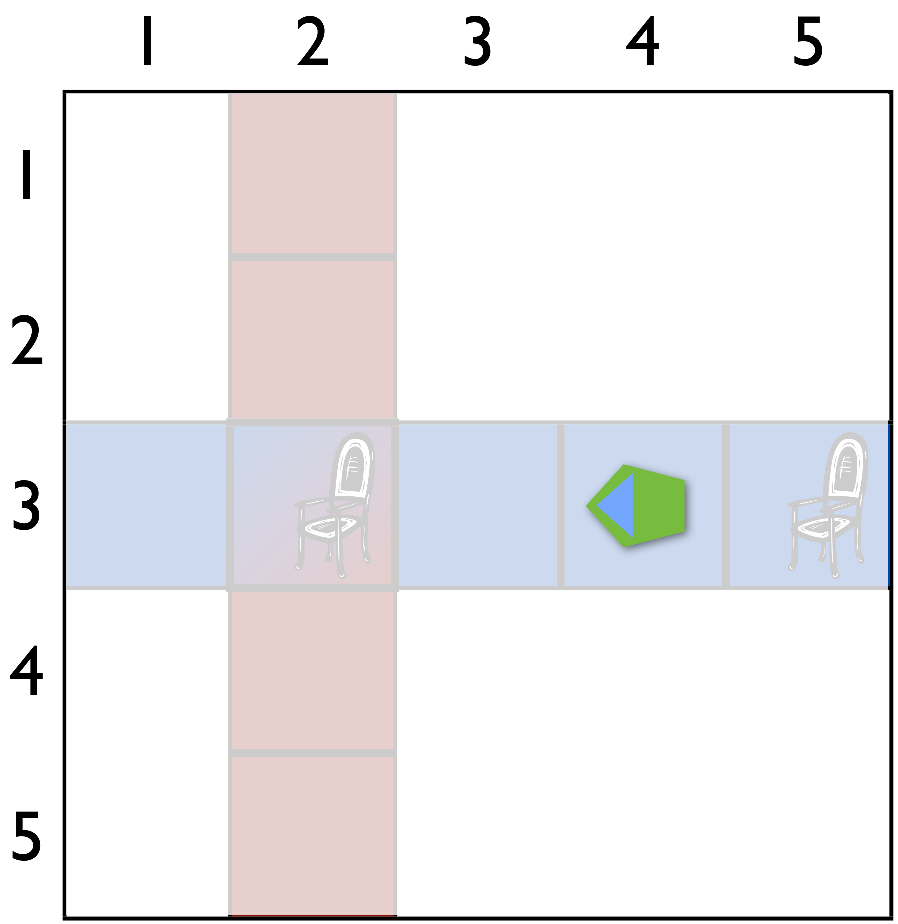


Noun phrases name specific entities

the chair

$\iota x.chair(x)$

Spatial Language



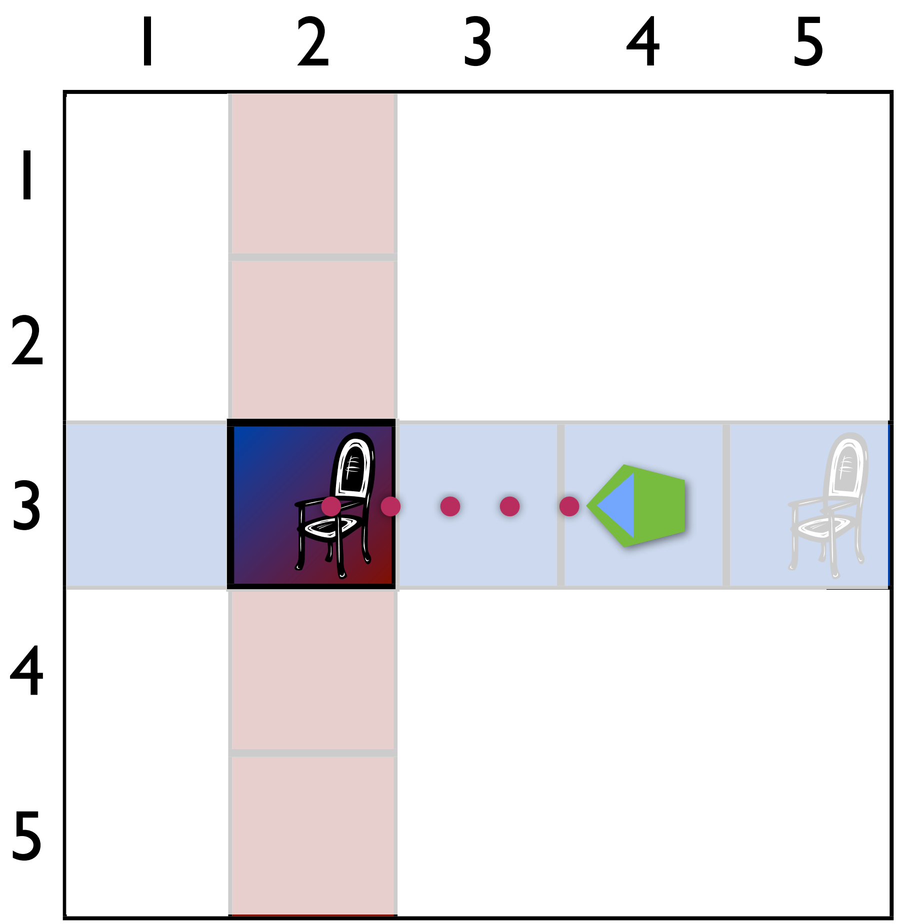
Noun phrases name specific entities

the chair

$\iota x.chair(x)$

Definite determiner depends on agent state

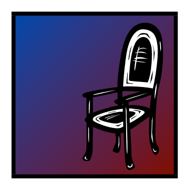
Spatial Language



Noun phrases name specific entities

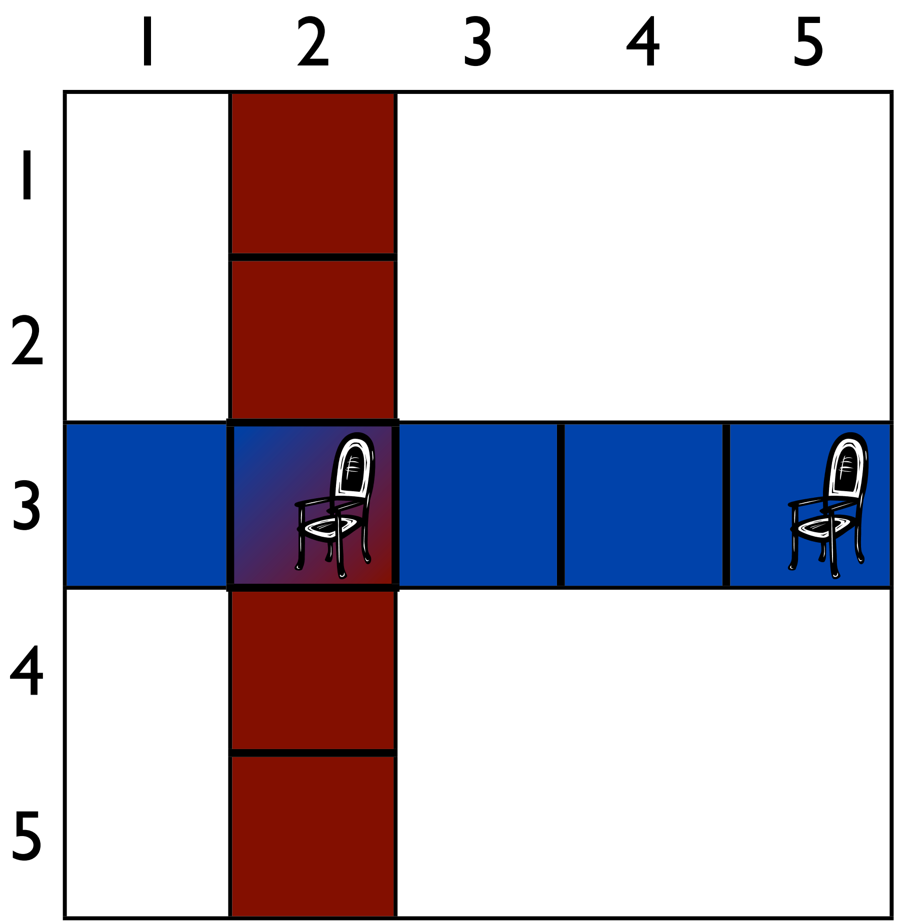
the chair

$\iota x.chair(x)$



Definite determiner depends on agent state

Spatial Language

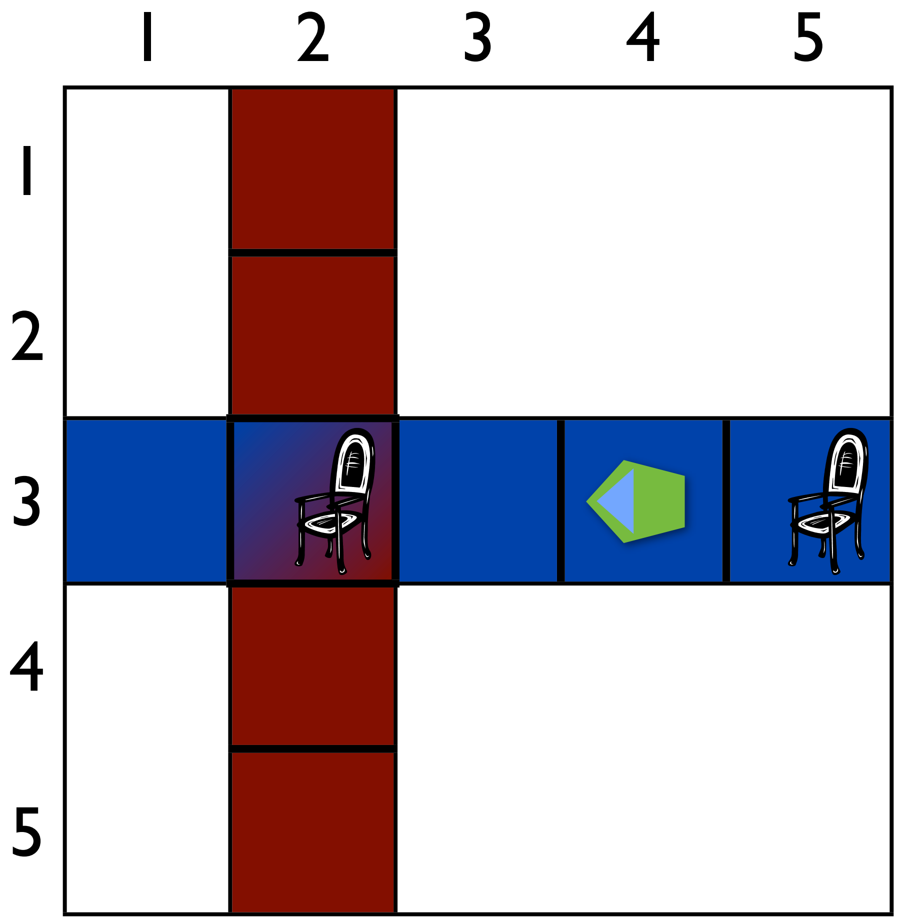


- Nouns
- Noun phrases
- Adjectives
- Prepositional phrases
- Spatial relations

Modeling Instructions

- Sequences of identical actions are events
- Use Neo-Davidsonian event semantics
- Represent imperatives as sets of events

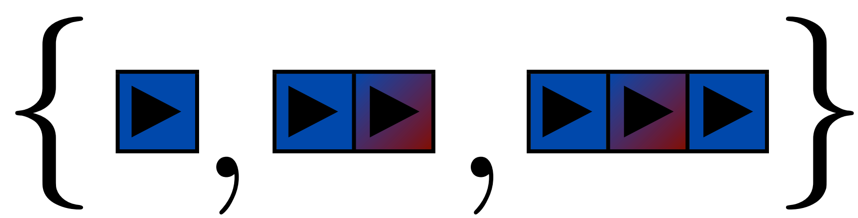
Modeling Instructions



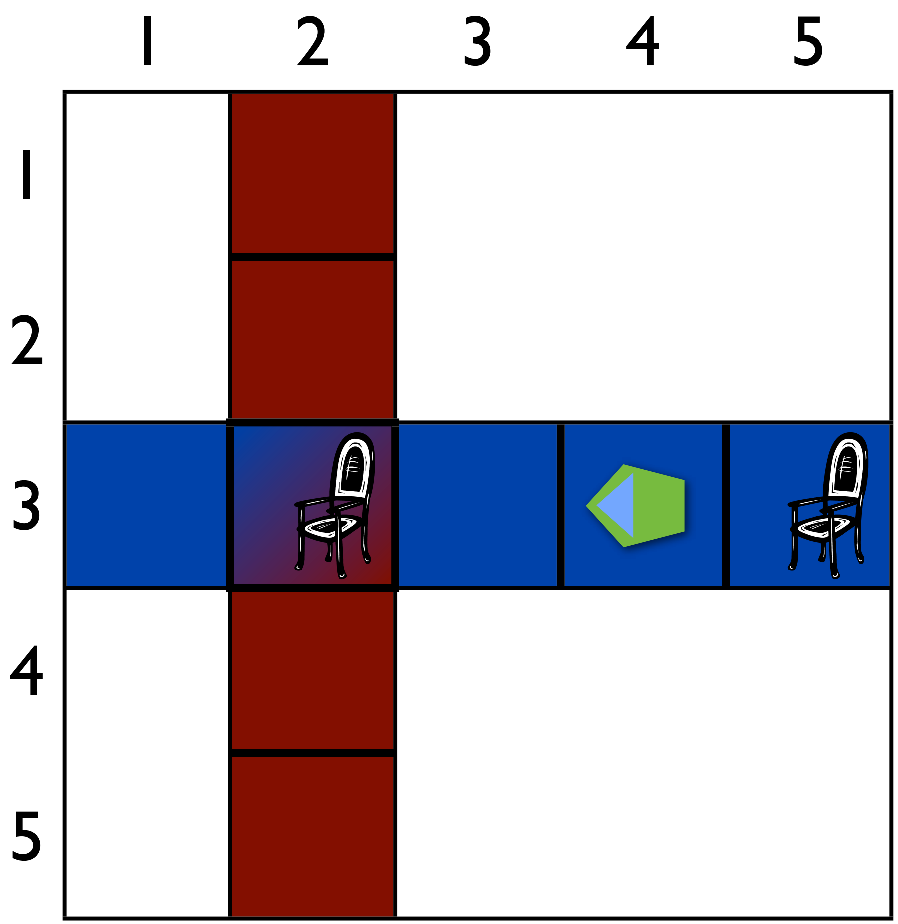
Imperatives are sets of events

move

$\lambda a.move(a)$



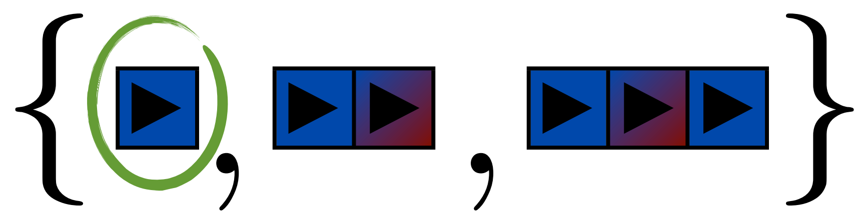
Modeling Instructions



Imperatives are sets of events

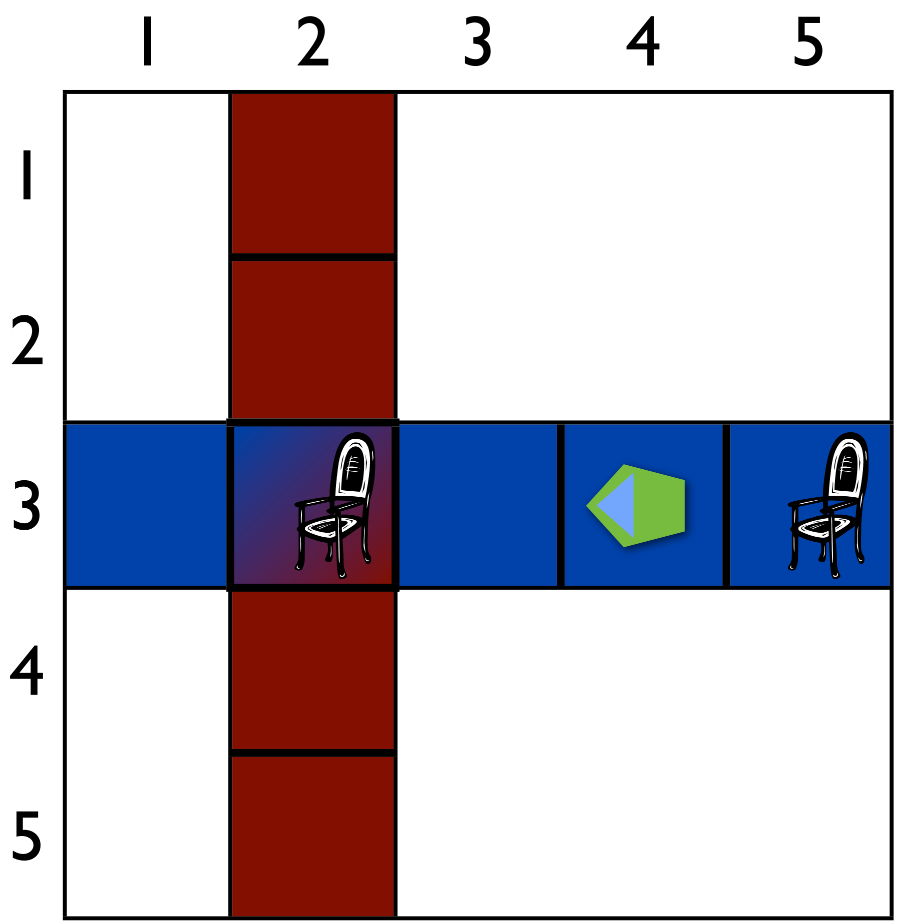
move

$\lambda a.move(a)$



Disambiguate by preferring shorter sequences

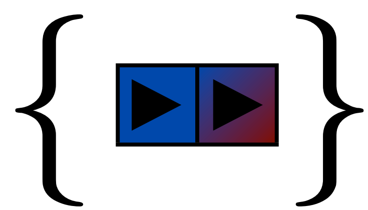
Modeling Instructions



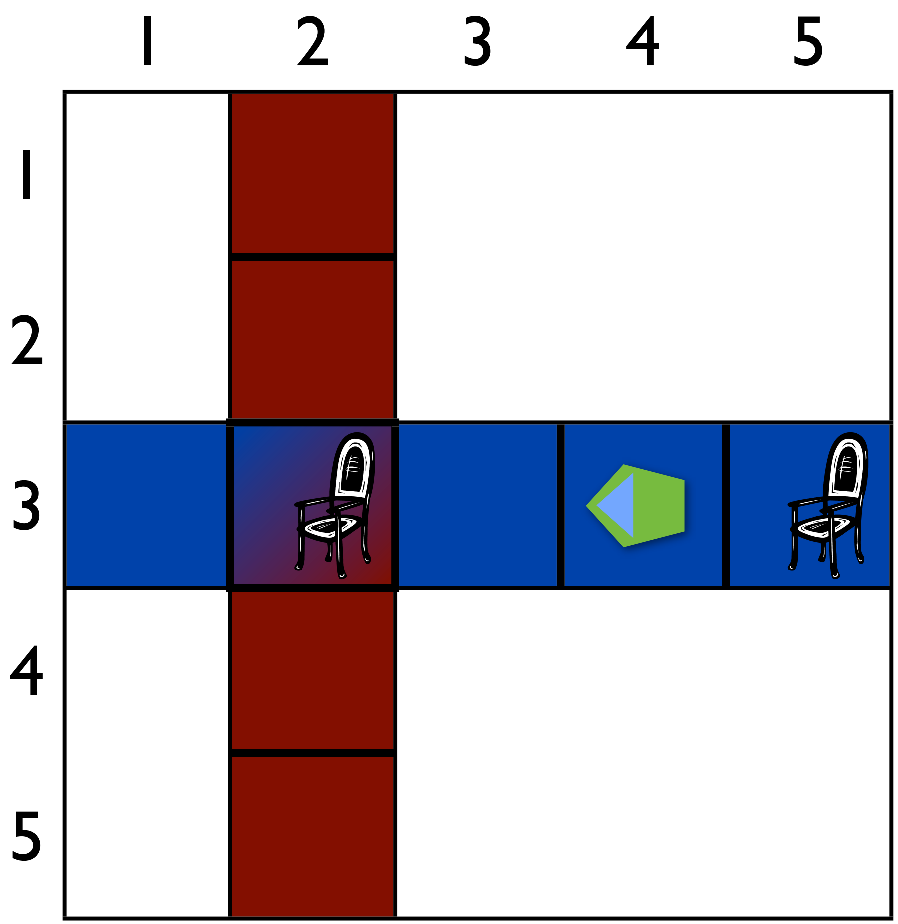
Events can be modified by adverbials

go to the chair

$$\lambda a.move(a) \wedge to(a, \iota x.chair(x))$$



Executing Logical Forms



go to the chair

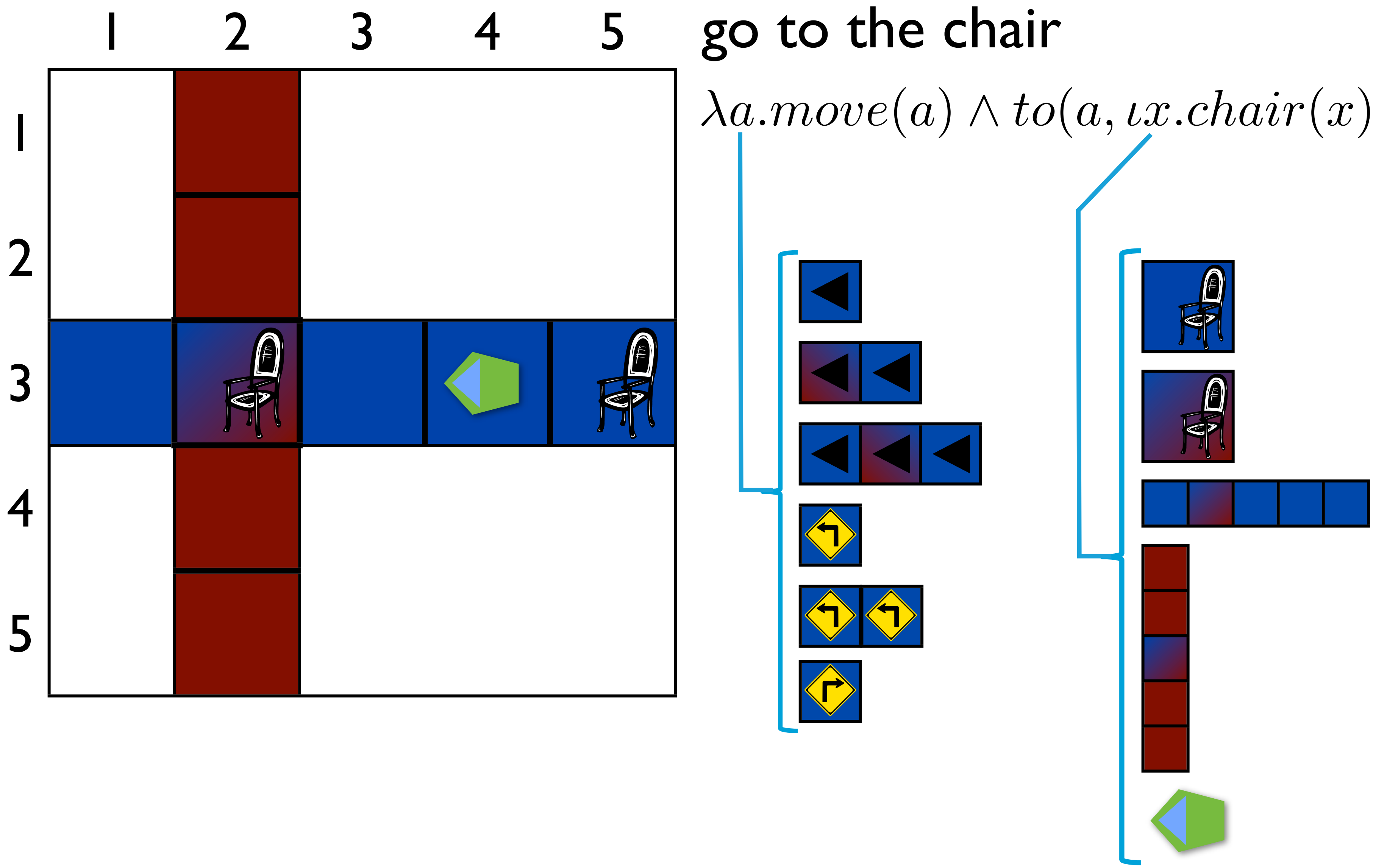
$\lambda a.move(a) \wedge to(a, \iota x.chair(x))$

Consider all variable assignments to find satisfying ones

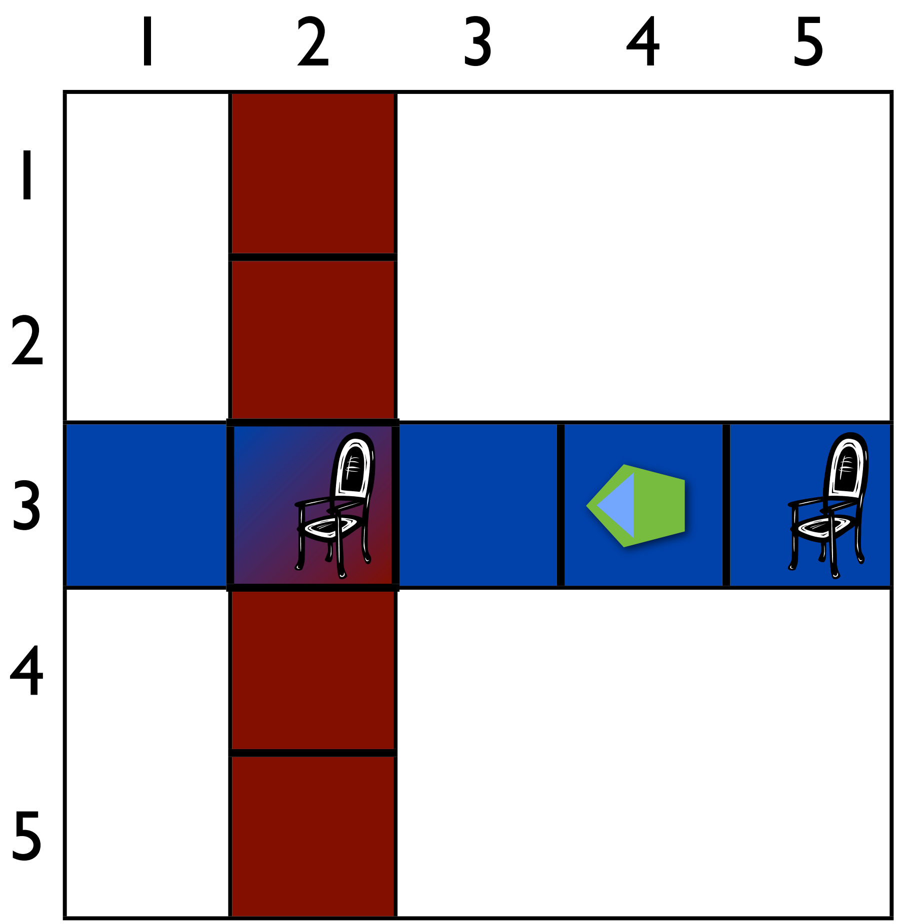
Executing Logical Forms

go to the chair

$$\lambda a.move(a) \wedge to(a, \iota x.chair(x))$$

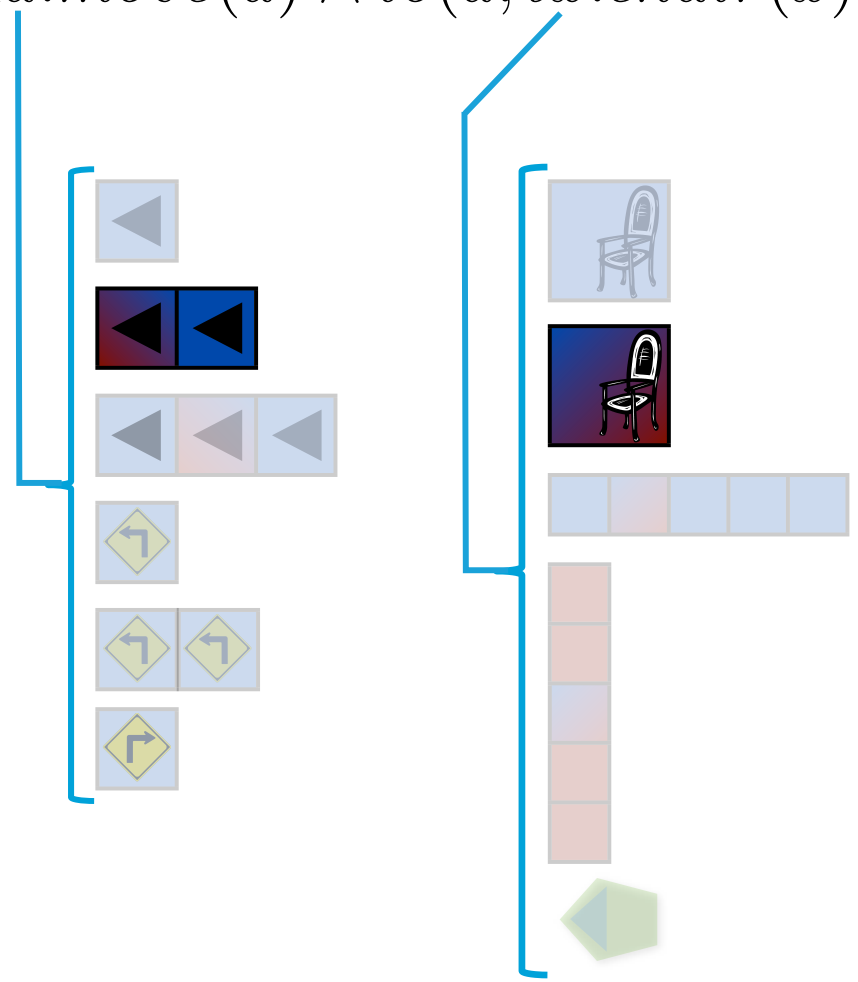


Executing Logical Forms

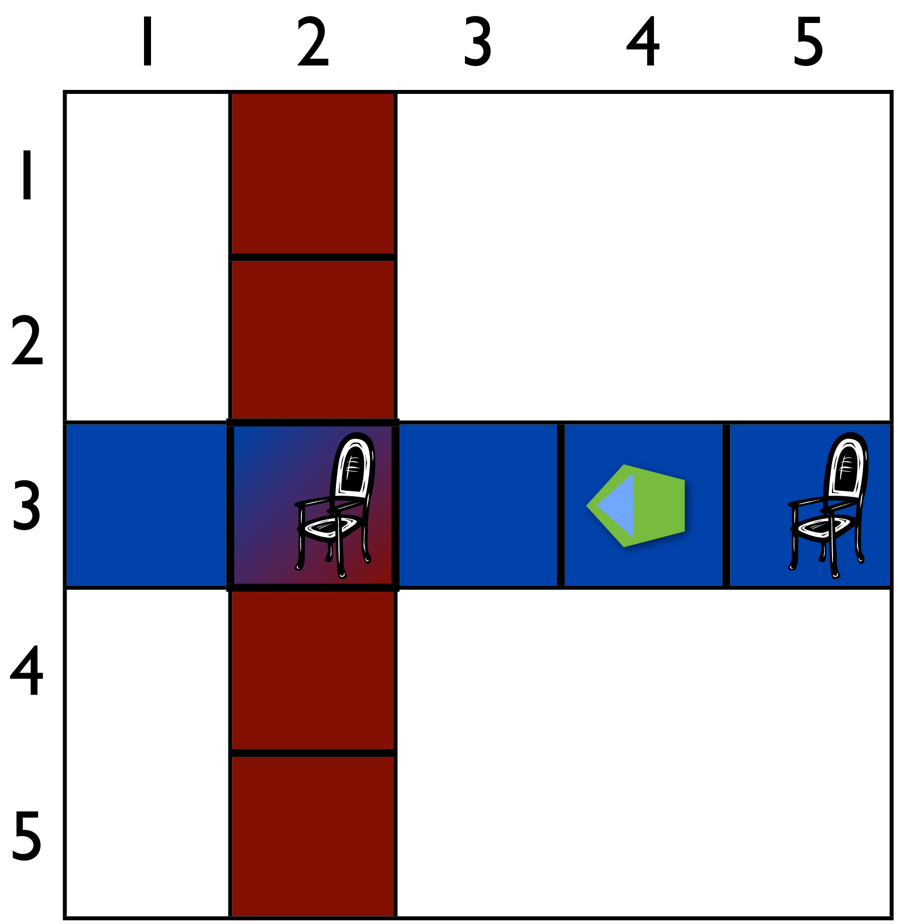


go to the chair

$\lambda a.move(a) \wedge to(a, \iota x.chair(x))$



Dynamic Models

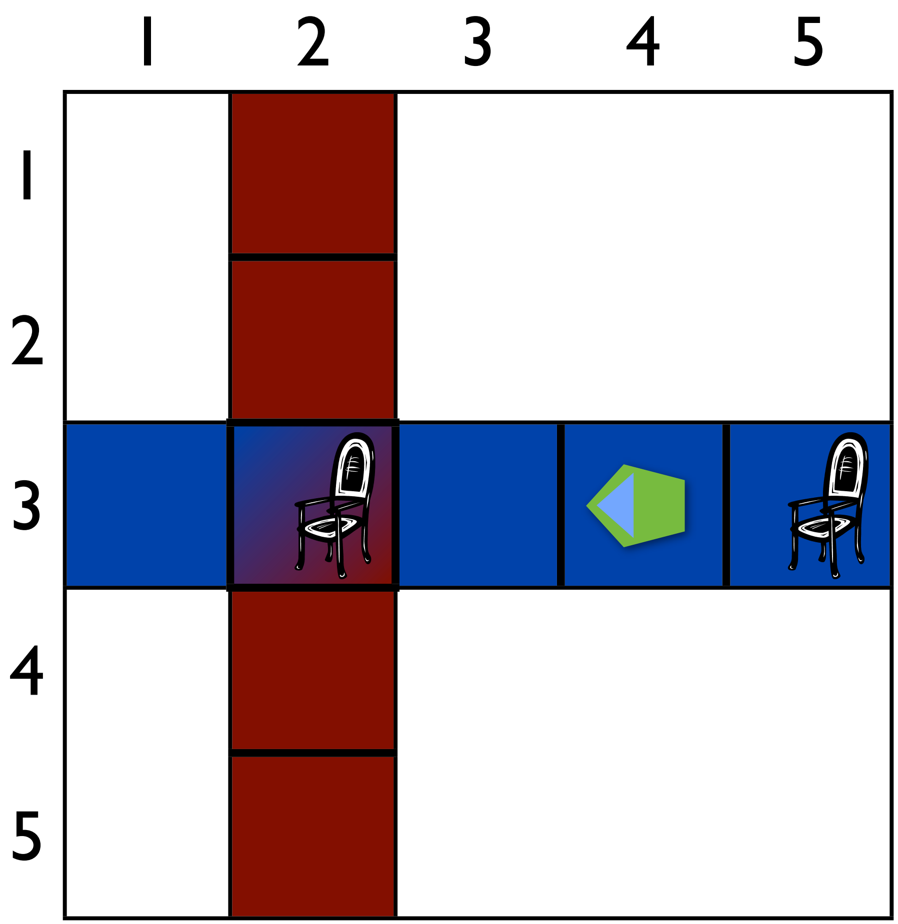


World models change during execution

move until you reach the chair

$$\lambda a.move(a) \wedge$$
$$post(a, intersect(\iota x.chair(x), you))$$

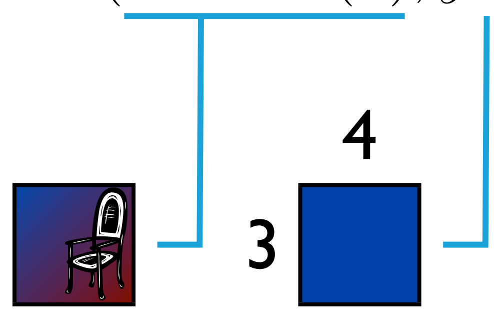
Dynamic Models



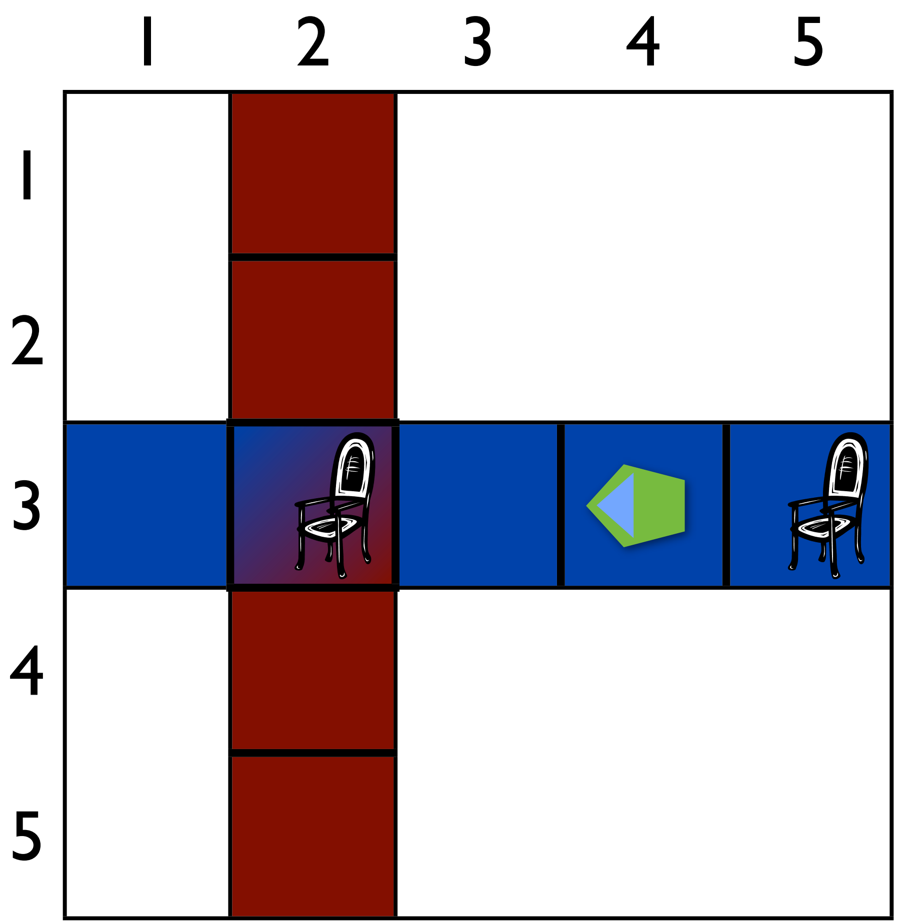
World models change during execution

move until you reach the chair

$$\lambda a.move(a) \wedge post(a, intersect(\iota x.chair(x), you))$$



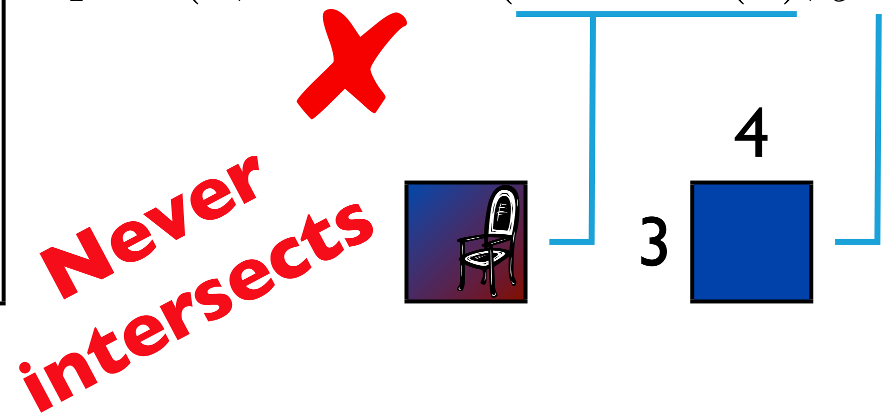
Dynamic Models



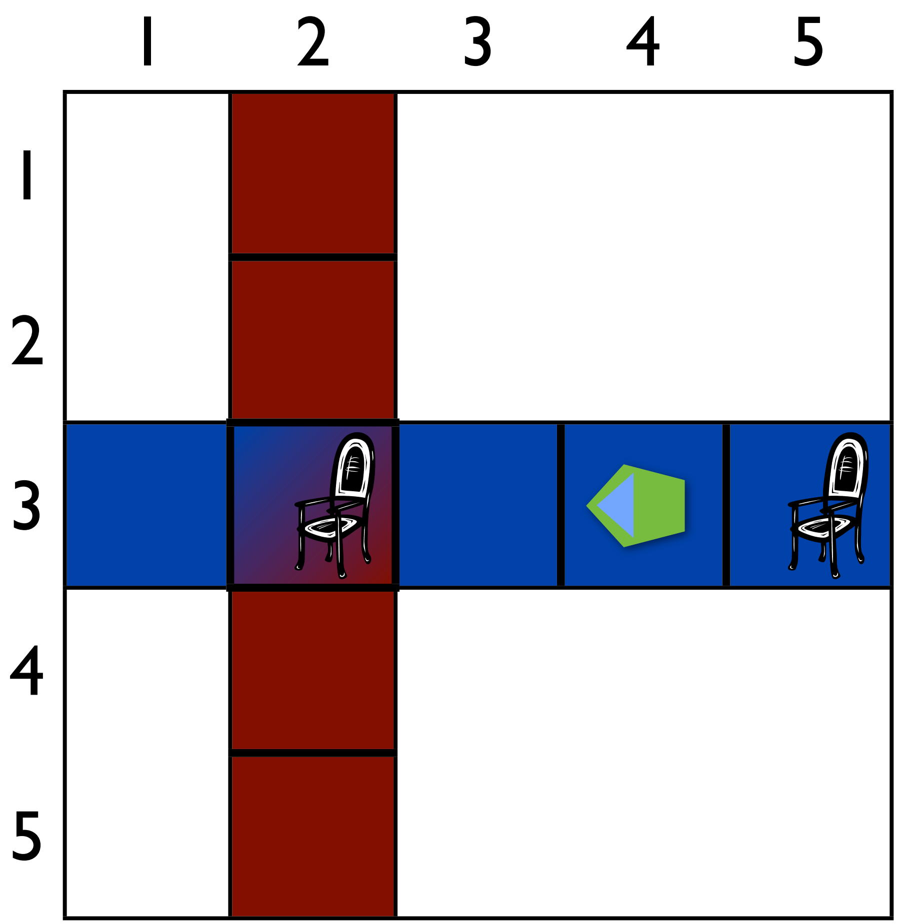
World models change during execution

move until you reach the chair

$$\lambda a.move(a) \wedge$$
$$post(a, intersect(\iota x.chair(x), you))$$



Dynamic Models



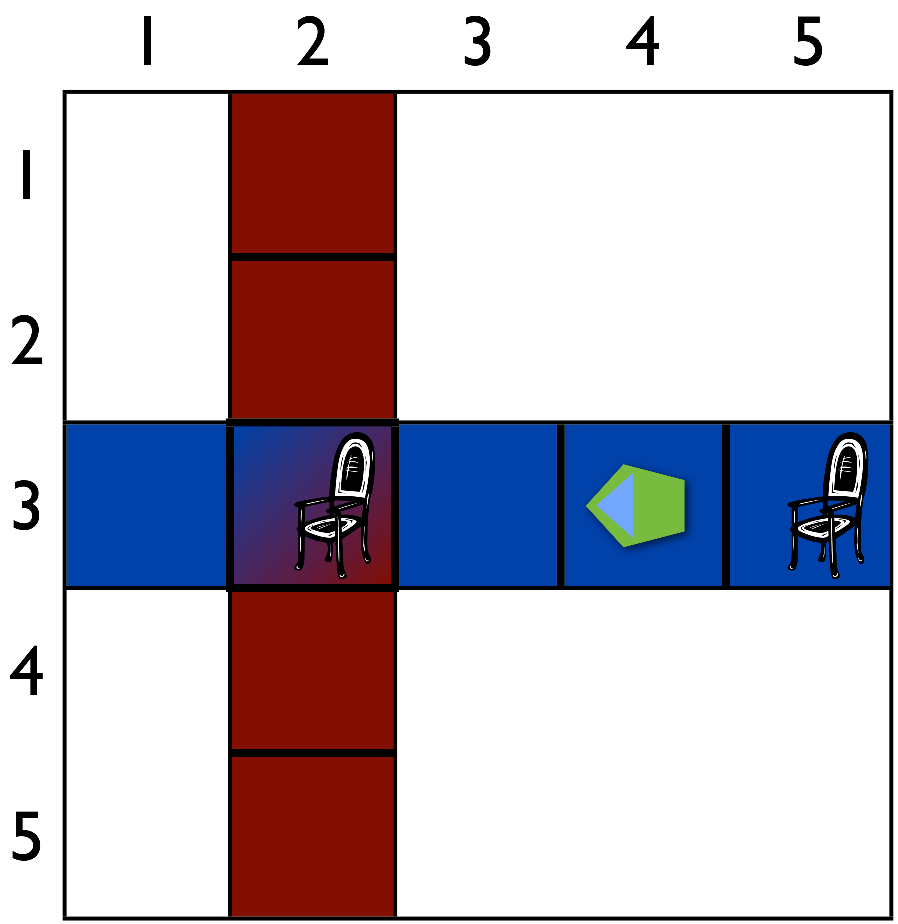
World models change during execution

move until you reach the chair

$$\lambda a.move(a) \wedge$$
$$post(a, intersect(\iota x.chair(x), you))$$

Update model to reflect state change

Dynamic Models



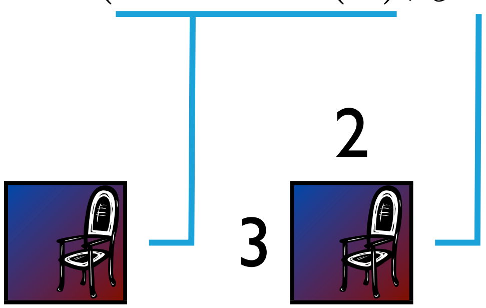
World models change during execution

move until you reach the chair

$\lambda a.move(a) \wedge$

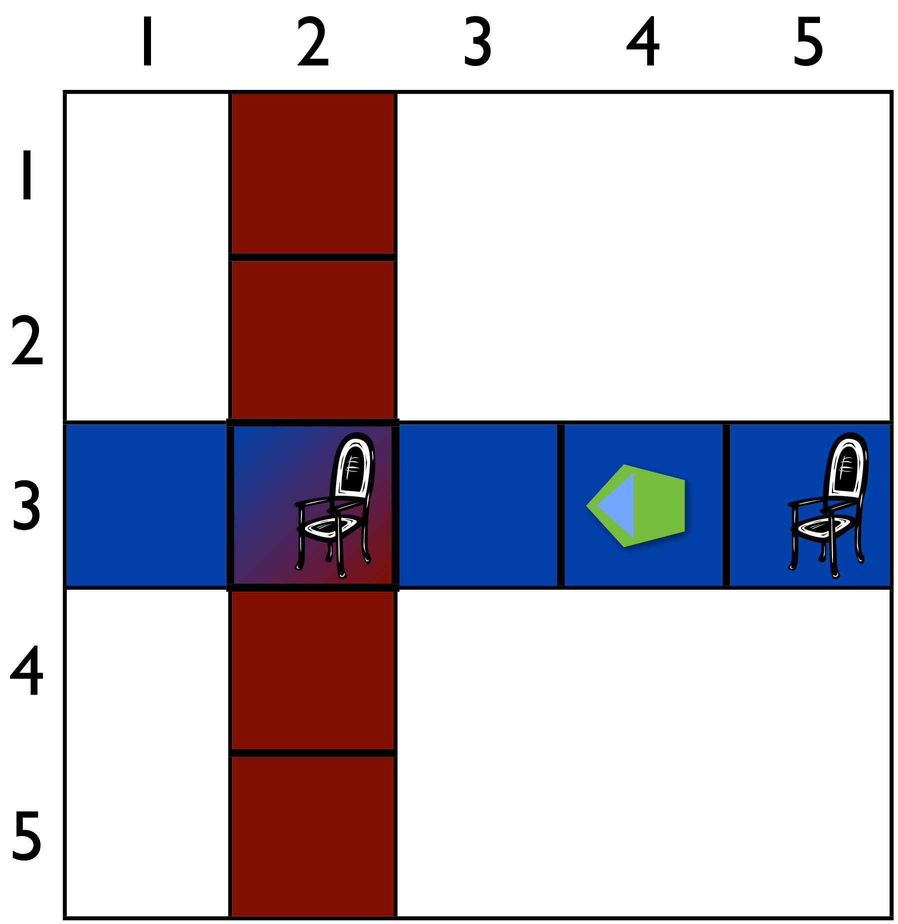
$post(a, intersect(\underbrace{\iota x.chair(x)}, \underbrace{you}))$

Update



Update model to reflect state change

Implicit Actions

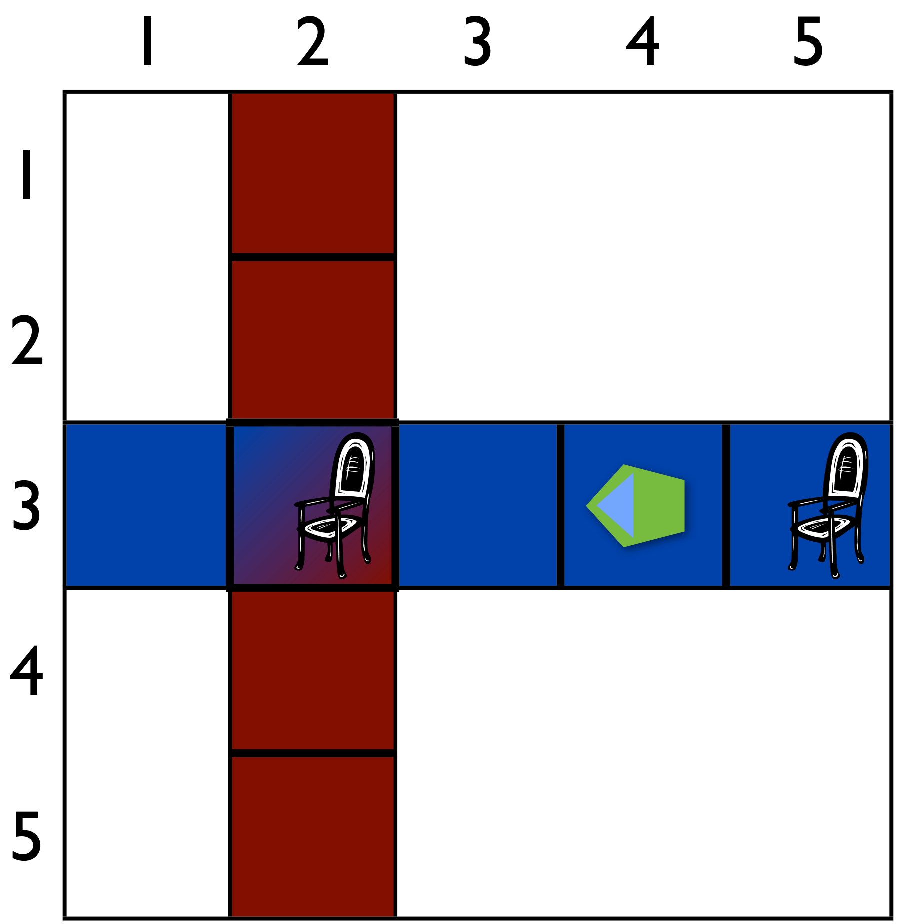


Consider actions with prefixed implicit actions

at the chair, turn left

$$\lambda a.turn(a) \wedge dir(a, left) \wedge$$
$$pre(a, intersect(\iota x.chair(x), you))$$

Implicit Actions

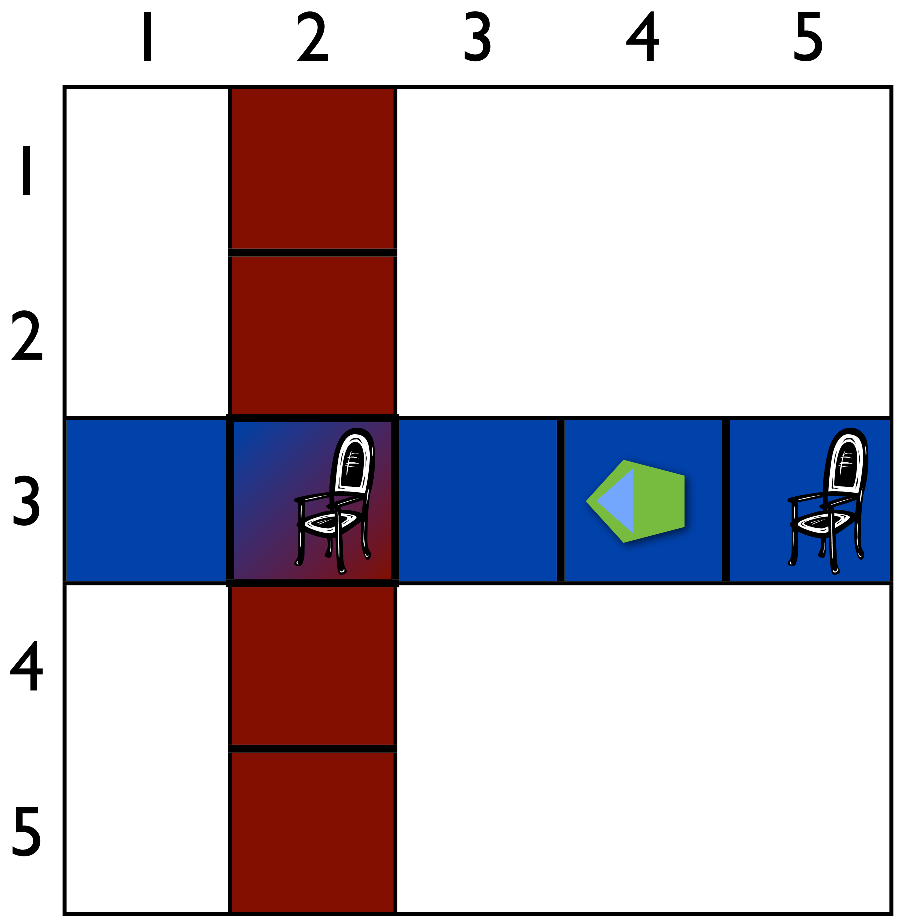


Consider actions with prefixed implicit actions

at the chair, turn left

$$\lambda a.turn(a) \wedge dir(a, left) \wedge pre(a, intersect(\iota x.chair(x), you))$$

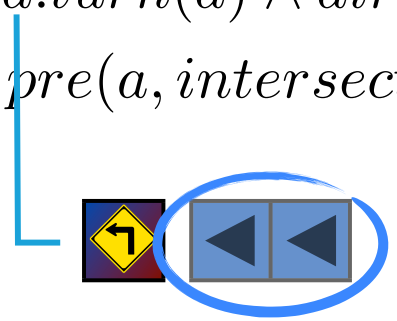
Implicit Actions



Consider actions with prefixed implicit actions

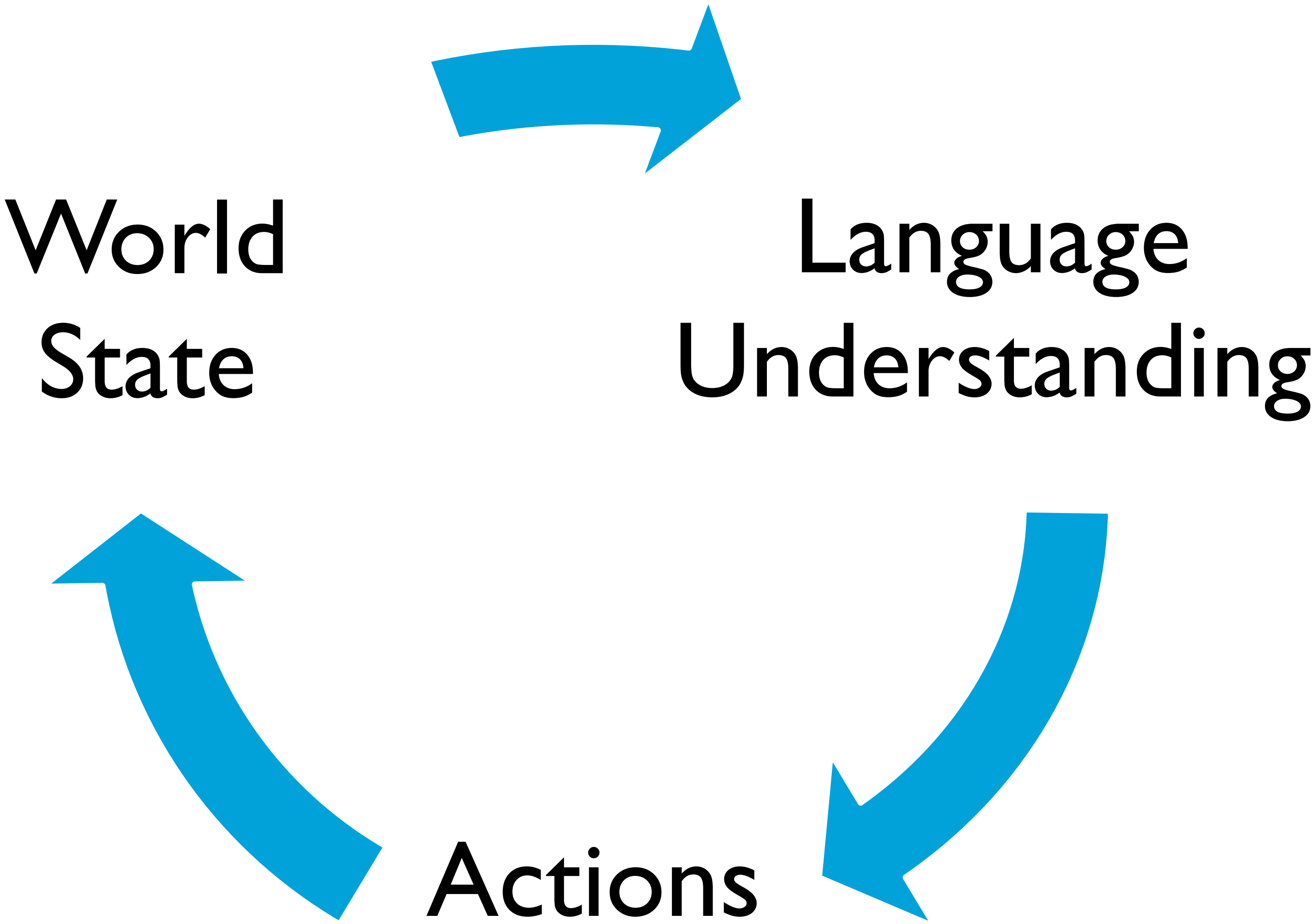
at the chair, turn left

$$\lambda a.turn(a) \land dir(a, left) \land pre(a, intersect(\iota x.chair(x), you))$$

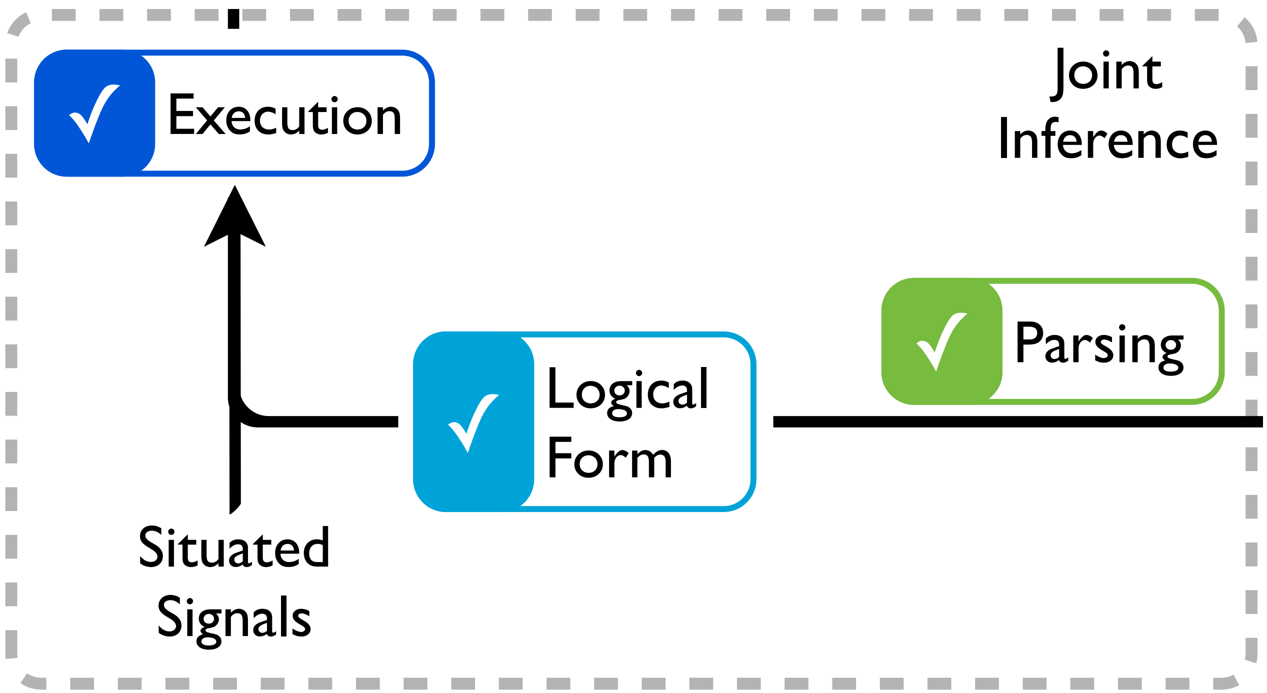


Implicit actions

Joint Inference

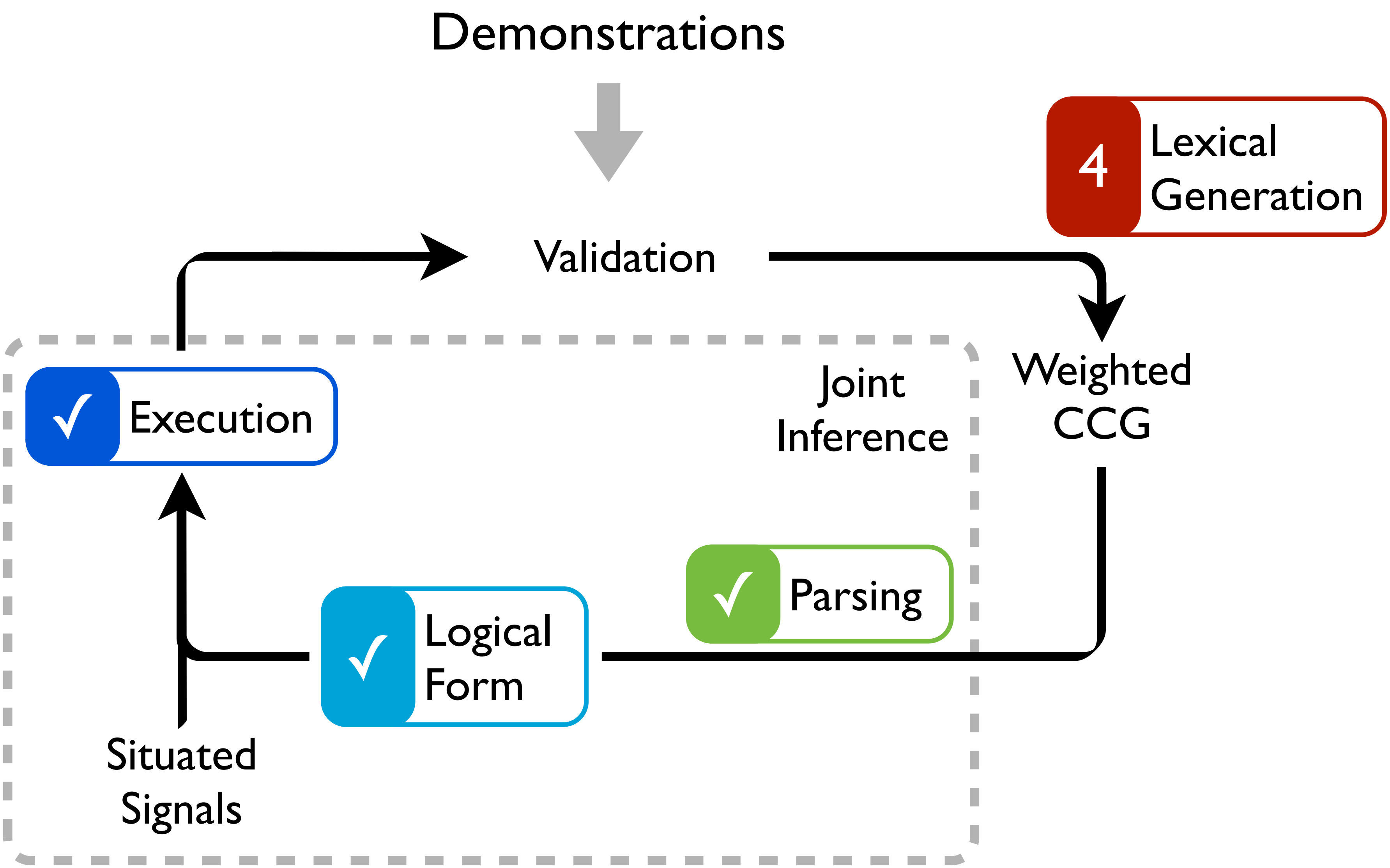


Learning



- Execution is integrated into parsing
- During parsing, execute logical forms and observe the result

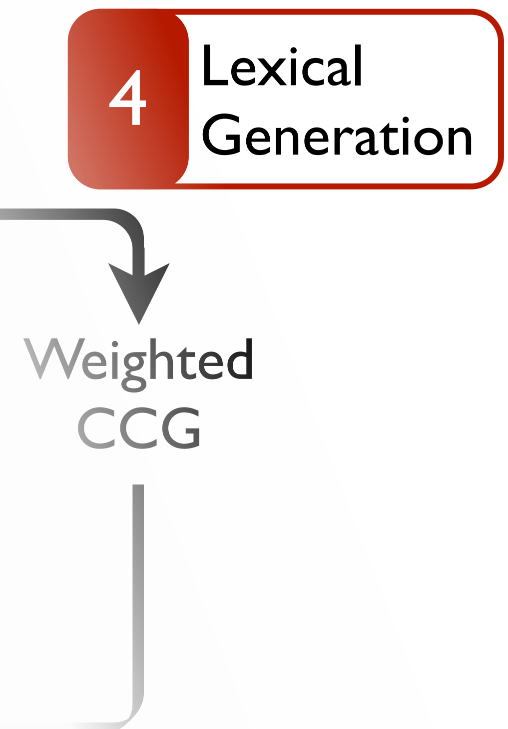
Learning



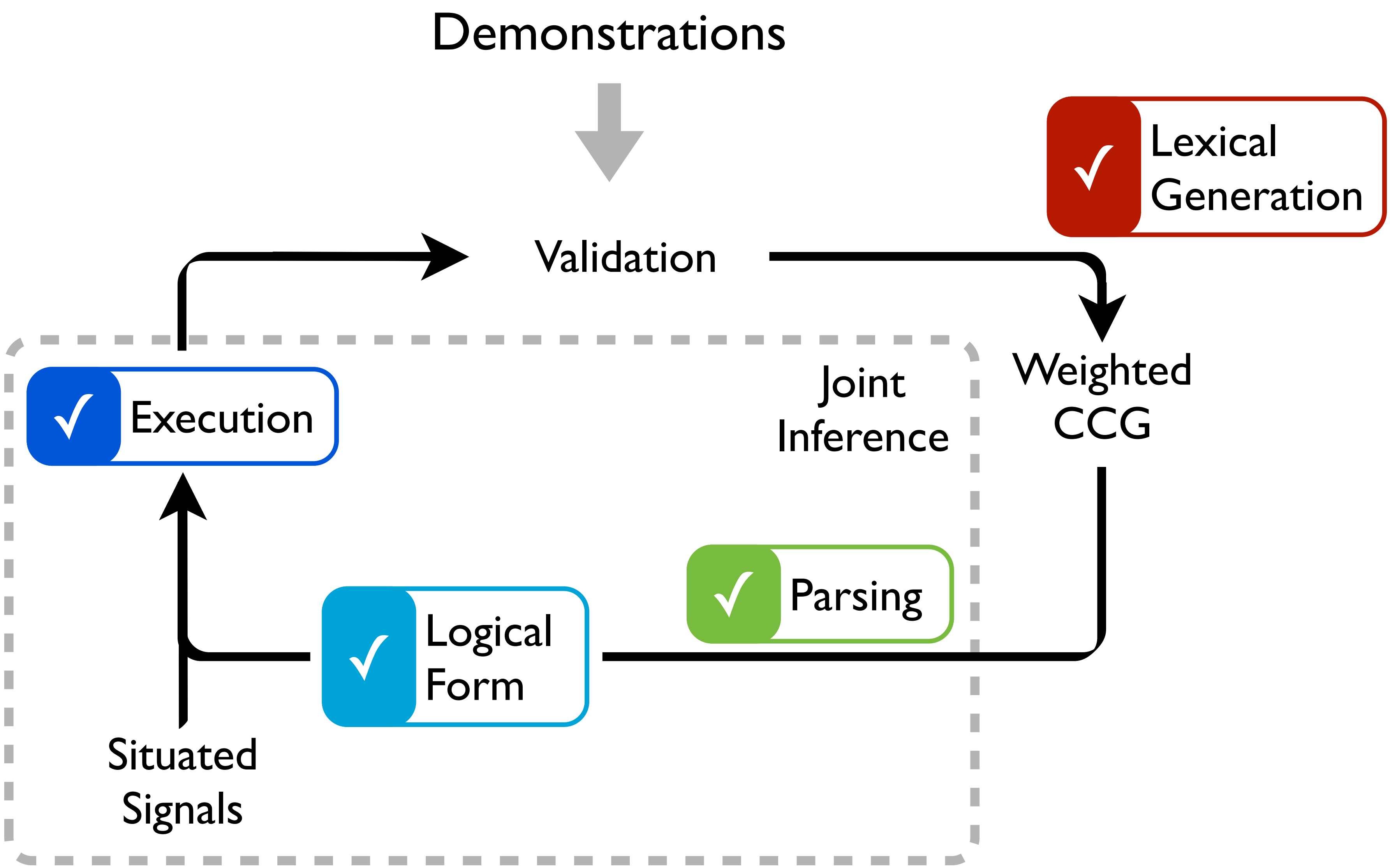
Demonstrations
Validation
4
Lexical
Generation
Weighted
CCG
Joint
Inference
√
Execution
√
Parsing
√
Logical
Form
Situated
Signals

Learning

- Based on a small set of seed templates
- New coarse-to-fine parsing algorithm to gradually prune the potential lexical entries
- Conservative approach to introduce new entries to the model



Learning


Demonstrations
Lexical Generation
Validation
Weighted CCG
Execution
Joint Inference
Parsing
Logical Form
Situated Signals

Validation-Driven Learning

- Online
- 2 steps:
  - Lexical generation
  - Parameter update
- Driven by a weak validation signal

Validation-Driven Learning

For T iterations, for each training sample:

- Step 1: Lexical generation
  - Generate a large number of potential lexical entries
  - Parse with the generated lexicon using the model
  - Select the best valid parses from the k-best parses
  - Add their lexical items to the lexicon
- Step 2: Update parameters

Validation-Driven Learning

For T iterations, for each training sample:

- Step 1: Lexical generation
- Step 2: Update parameters
  - Parse using the model
  - Split all parses into two sets:
    max scoring valid and invalid
  - Find margin violating pairs between the sets
  - Do a perceptron-style update using these violations

Related Work

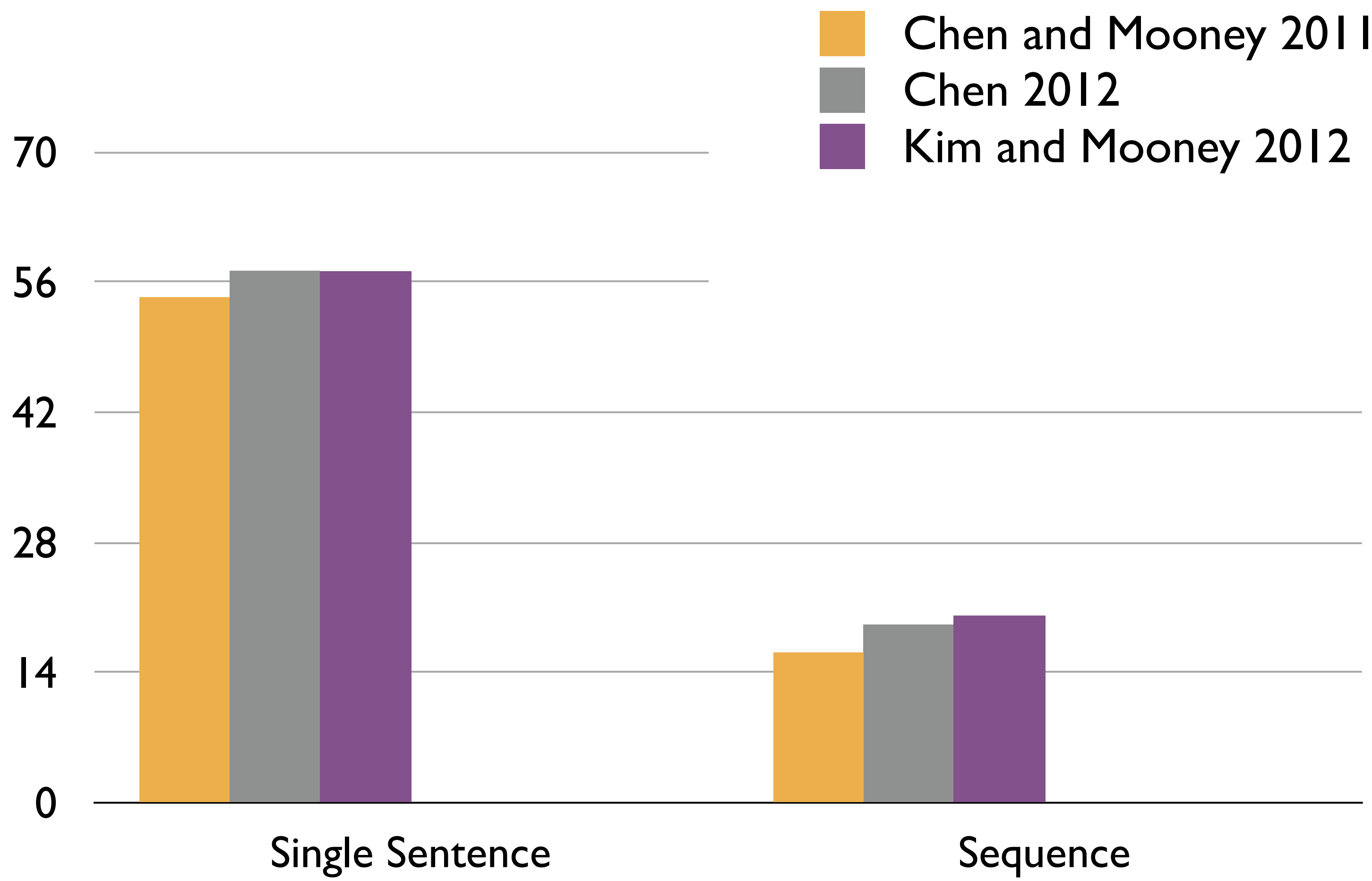
Supervised semantic parsing	[Kate, Mooney 2006; Wong, Mooney 2007; Muresan 2011]
with CCGs	[Zettlemoyer, Collins 2005; 2007; Kwiatkowski et al. 2010; 2011]
Weakly supervised semantic parsing	[Clarke et al. 2010; Goldwasser, Roth 2011; Liang et al. 2011; Kirshnamurthy, Mitchell 2012; Goldwasser et al. 2011]
Grounded Semantic Analysis	[Liang et al. 2009; Chen et al. 2010; Matuszek et al. 2012]
Executing Instructions with Shallow Representation	[Branavan et al. 2009; 2010; Vogel, Jurafsky 2010; Wei et al. 2009; Kollar et al. 2010; Tellex et al. 2011]
Non-joint Execution of Instructions	[Matuszek et al. 2010; 2012; Chen, Mooney 2011; Chen 2012; Kim, Mooney 2012]

Experimental Setup

- Seed lexicon from an annotated randomly selected 12 instruction sequences
- Features: lexical, type-raising usage and repetitions in logical coordinations
- Consider only executable parses as complete

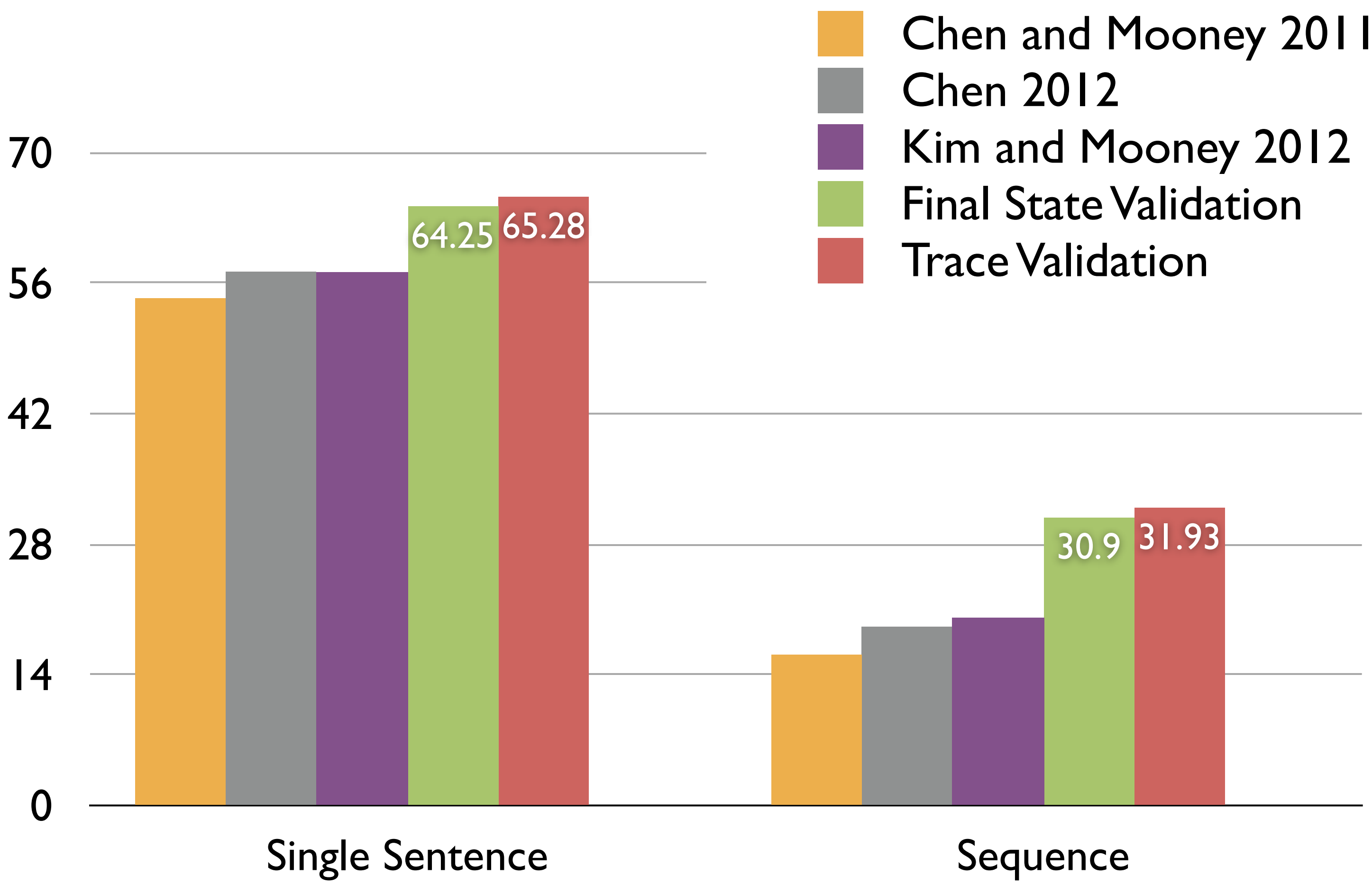
Results

SAIL Corpus - Cross Validation

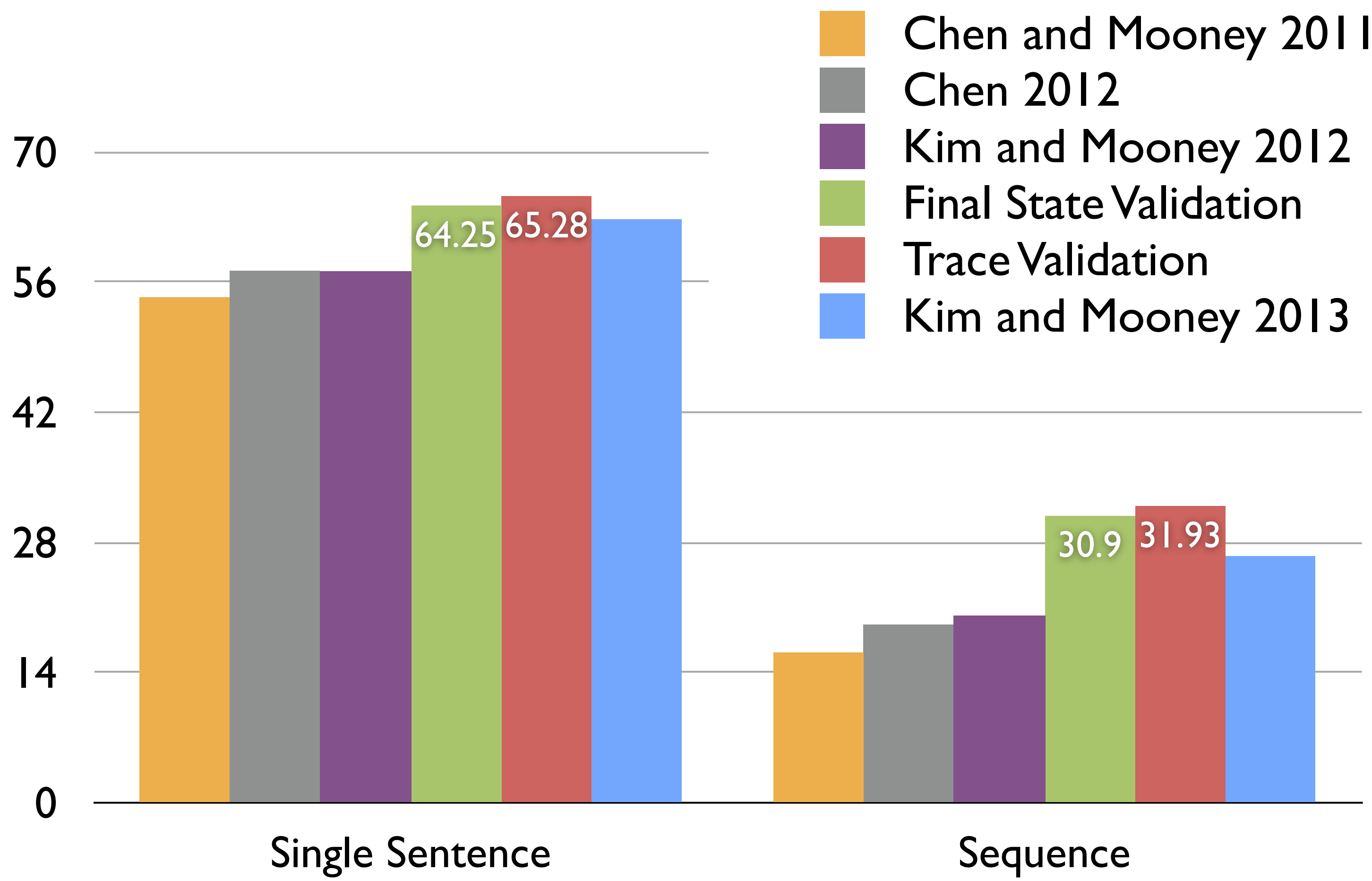


Results

SAIL Corpus - Cross Validation

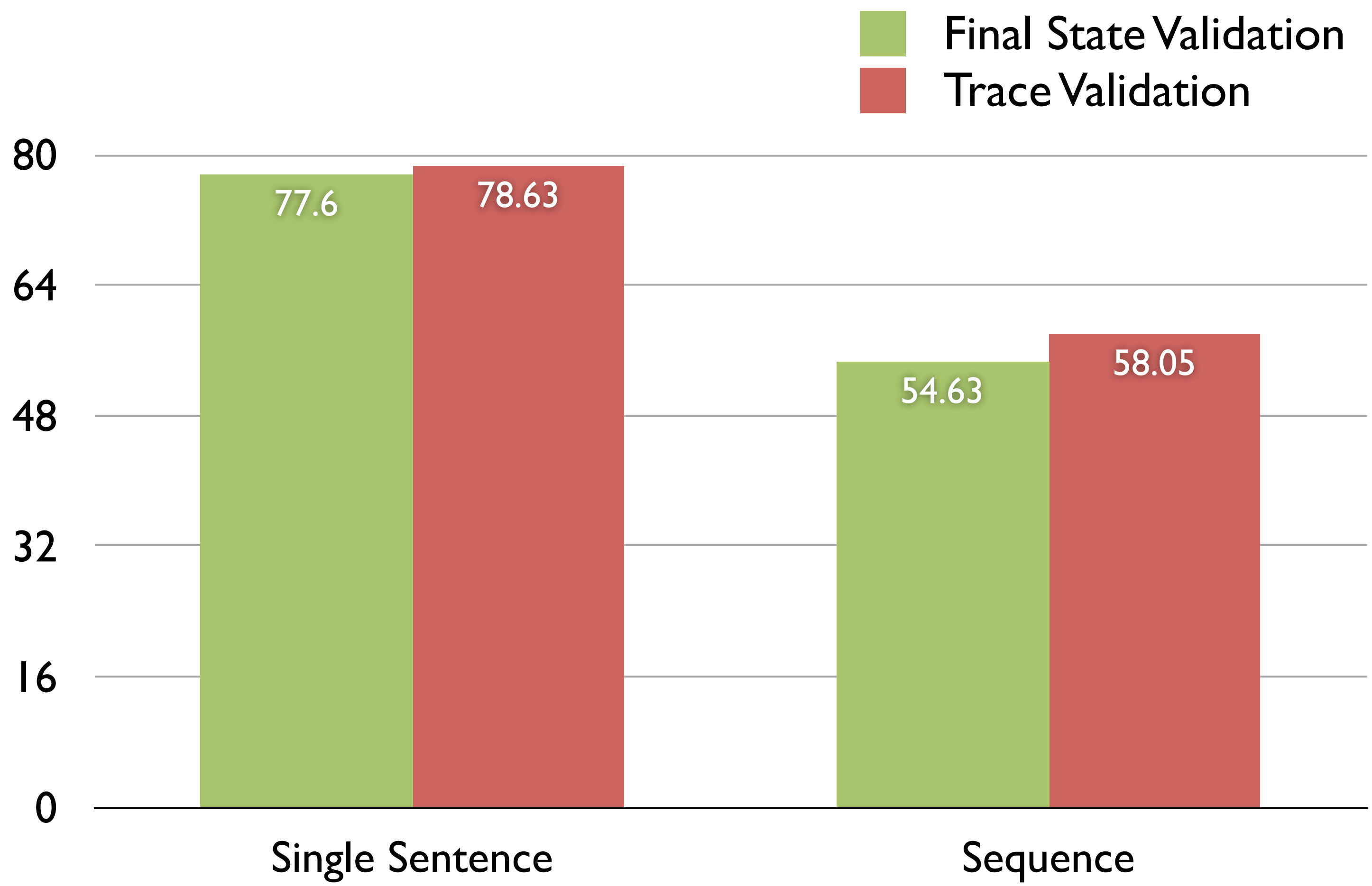


Results
SAIL Corpus - Cross Validation



Results

Oracle Corpus - Held-out Set



Contributions

- Linguistically-driven modeling of instructional language
- Joint inference for interpretation and execution of grounded language
- General weakly-supervised learning approach for semantic parsers

UW SPF

Open source semantic parsing framework

http://yoavartzi.com/spf

Semantic Parser	Flexible High-Order Logic Representation	Learning Algorithms

UW SPF

Open source semantic parsing framework

http://yoavartzi.com/spf

Semantic Parser	Flexible High-Order Logic Representation	Learning Algorithms

Navigation code and data available online

http://yoavartzi.com/navi

Coming up:
ACL tutorial

[fin]