Reading Between the Lines:
Learning to Map High-level Instructions to Commands

S.R.K. Branavan

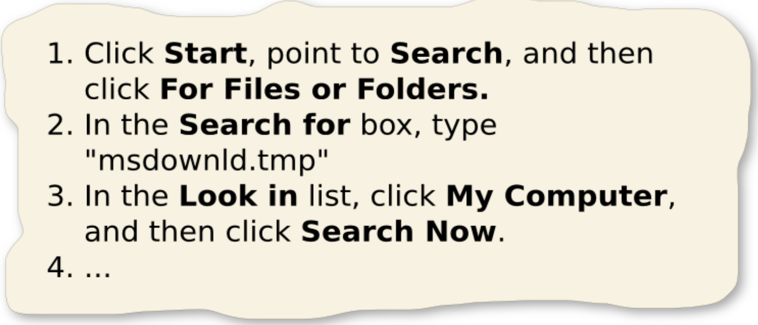
Joint work with: Luke Zettlemoyer, Regina Barzilay

MIT

Mapping Instructions to Commands

Input

Instructions:
text descriptions
of commands



1. Click **Start**, point to **Search**, and then click **For Files or Folders.**
2. In the **Search for** box, type "msdownld.tmp"
3. In the **Look in** list, click **My Computer**, and then click **Search Now**.
4. ...

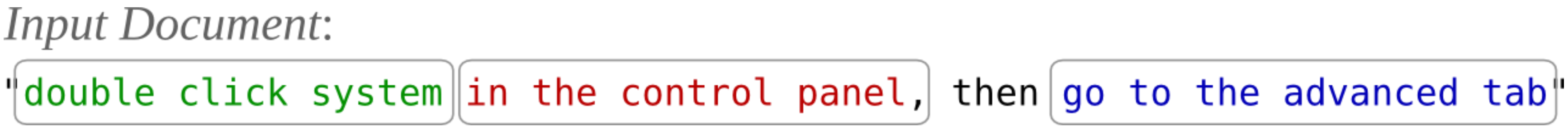
Target environment:
where commands
need to be executed

Output

Command sequence
executable in the
environment

```
LEFT_CLICK( Start )
LEFT_CLICK( Search )
  . . .
TYPE_INTO( Search for: ,"msdownld.tmp")
  . . .
```

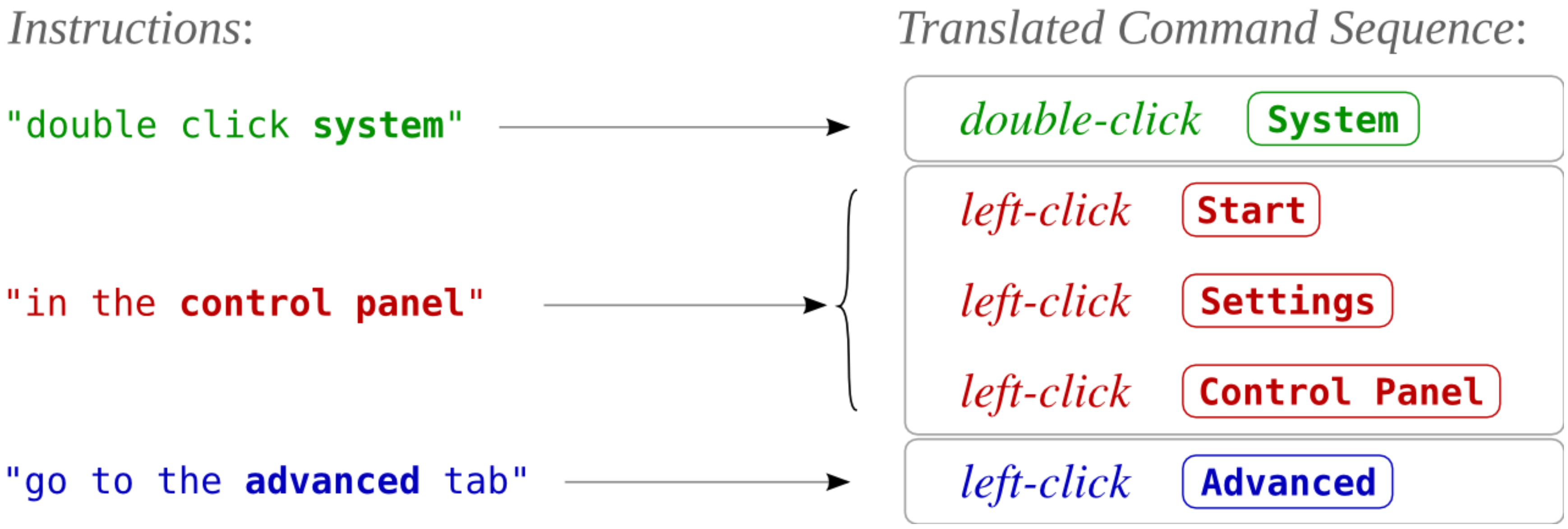
Segment Text into Individual Instructions



Input Document:

"double click system in the control panel, then go to the advanced tab"

Translate Instructions to Command Sequences



Instructions:

"double click **system**"

"in the **control panel**"

"go to the **advanced** tab"

Translated Command Sequence:

double-click **System**

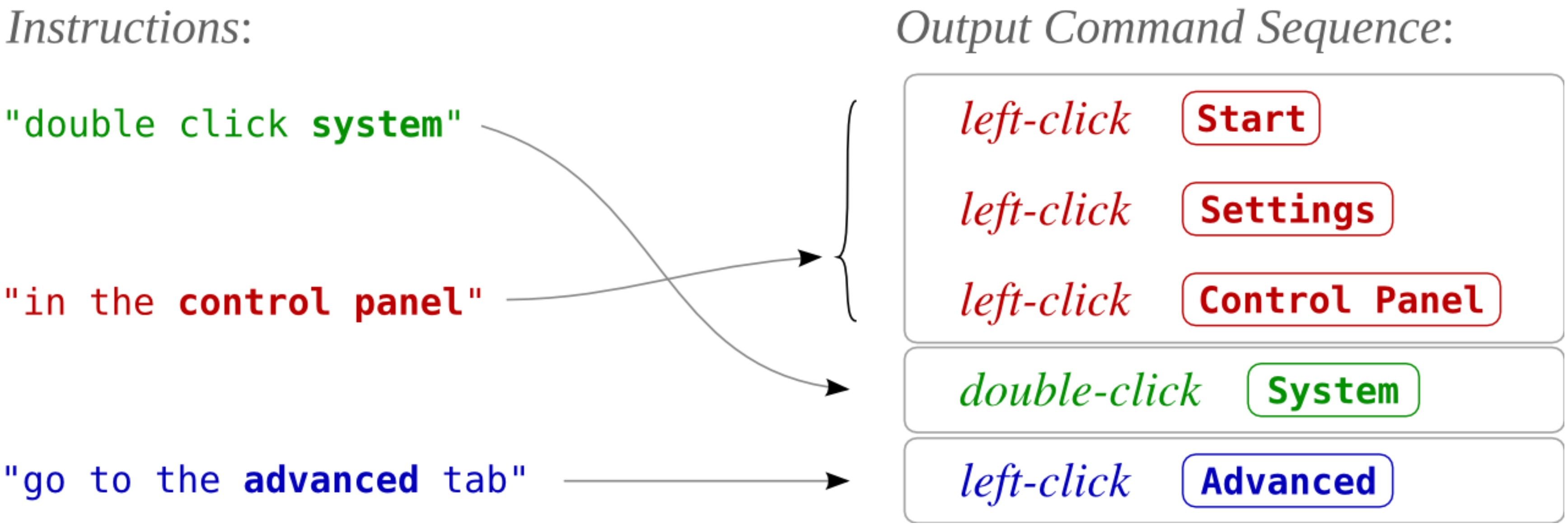
left-click **Start**

left-click **Settings**

left-click **Control Panel**

left-click **Advanced**

Permute Command Sequence to Execution Order



Instructions:

"double click **system**"

"in the **control panel**"

"go to the **advanced** tab"

Output Command Sequence:

left-click **Start**

left-click **Settings**

left-click **Control Panel**

double-click **System**

left-click **Advanced**

Previous Work: Low-level Instructions

(ACL 2009)

All commands are explicitly specified

Instructions	*Environment Commands*
"click start"	*left-click* **Start**
"click settings"	*left-click* **Settings**
"click control panel"	*left-click* **Control Panel**
"double click administrative tools"	*double-click* **Administrative Tools**
"double click computer management"	*double-click* **Computer Management**
"double click device manager"	*double-click* **Device Manager**
"under disk drives on the device list,"	*left-click* **disk drives**
"select the primary IDE drive"	*left-click* **Primay HDD**

Current Focus: High-level Instructions

Most commands are specified only implicitly

Instructions

"open device manager"

"under disk drives on the device list,"

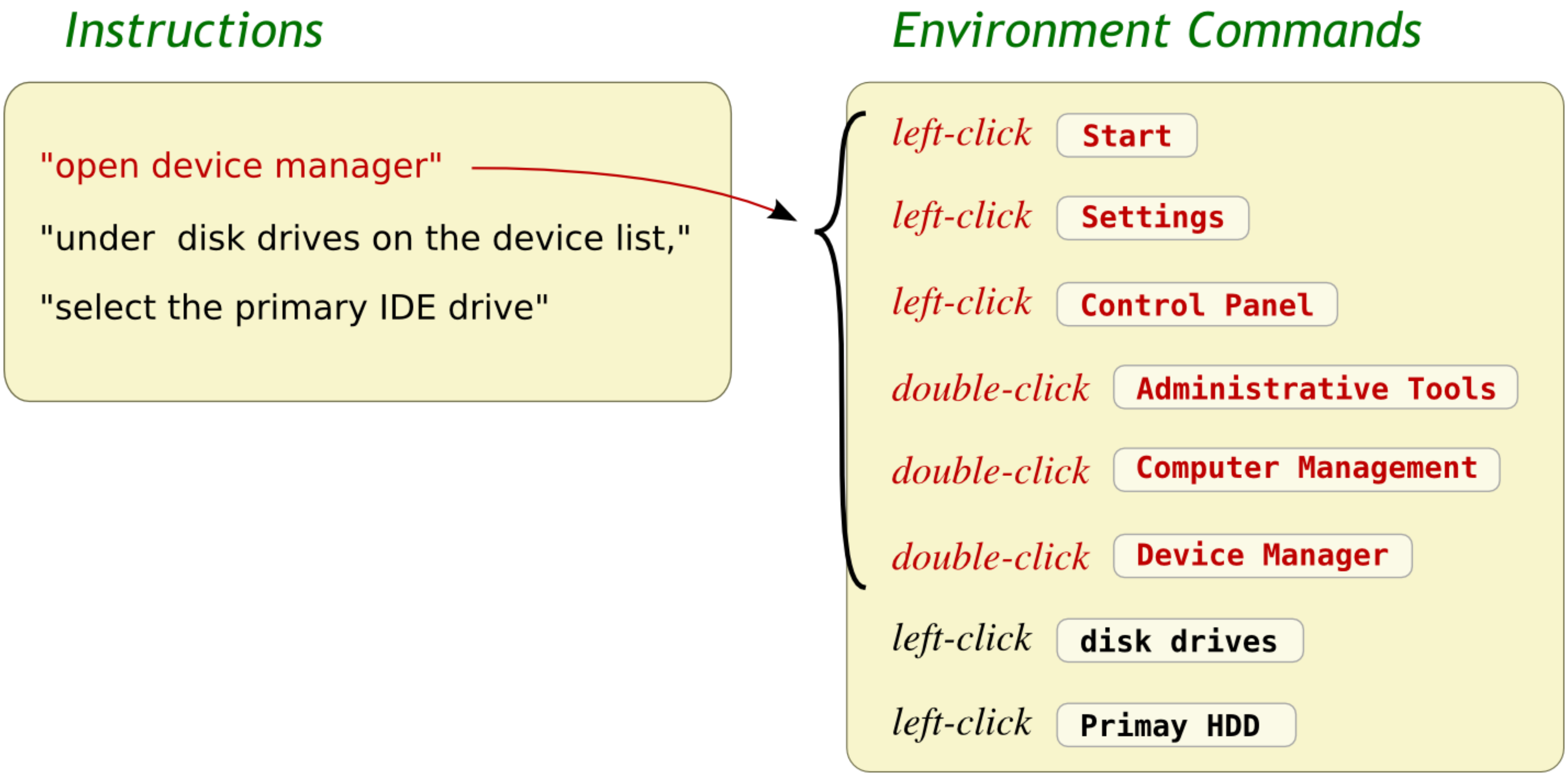
"select the primary IDE drive"

Environment Commands

left-click	**`Start`**
left-click	**`Settings`**
left-click	**`Control Panel`**
double-click	**`Administrative Tools`**
double-click	**`Computer Management`**
double-click	**`Device Manager`**
left-click	**`disk drives`**
left-click	**`Primay HDD`**

Current Focus: High-level Instructions

Most commands are specified only implicitly



Instructions

"open device manager"

"under disk drives on the device list,"

"select the primary IDE drive"

Environment Commands

left-click **Start**

left-click **Settings**

left-click **Control Panel**

double-click **Administrative Tools**

double-click **Computer Management**

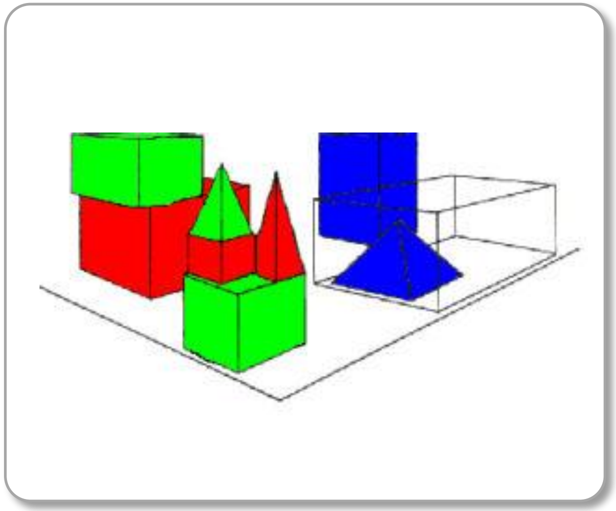
double-click **Device Manager**

left-click **disk drives**

left-click **Primay HDD**

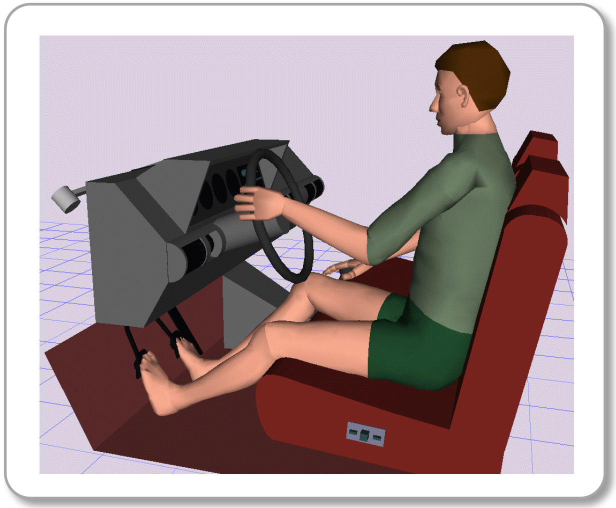
Interpreting High-level Instructions: An Old AI Problem

Winograd, 1972



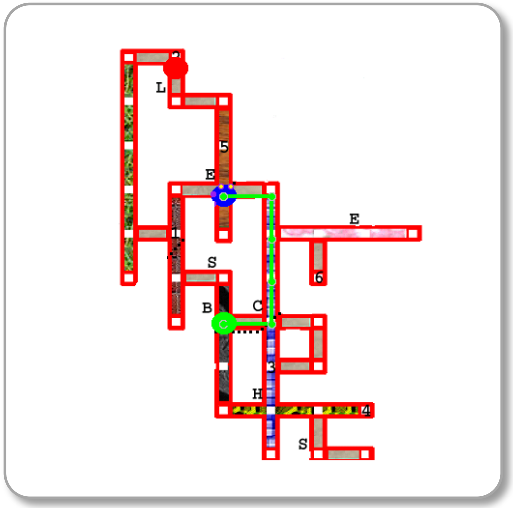
Block-worlds

Di Eugenio, 1992;
Webber, 1995



Virtual character animation

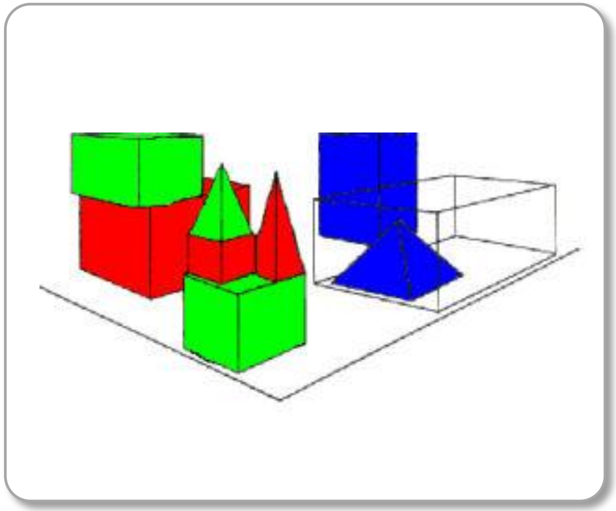
MacMahon, 2006



Navigation

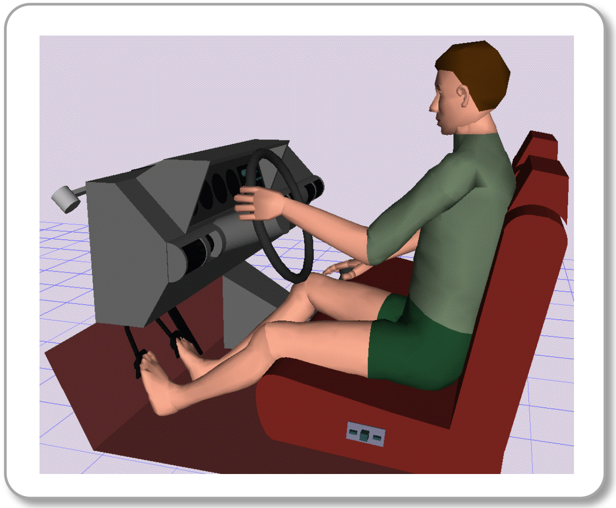
Interpreting High-level Instructions: An Old AI Problem

Winograd, 1972



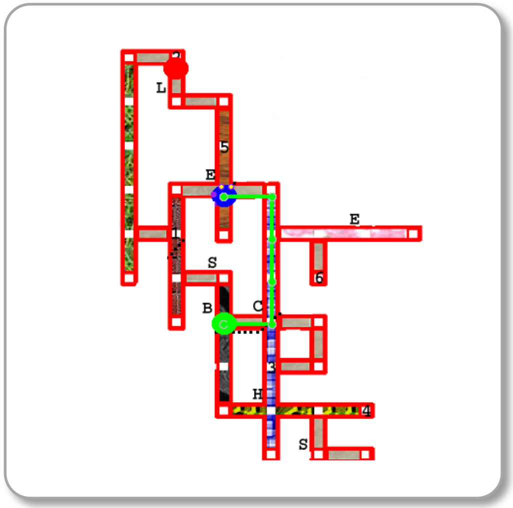
Block-worlds

Di Eugenio, 1992; Webber, 1995



Virtual character animation

MacMahon, 2006

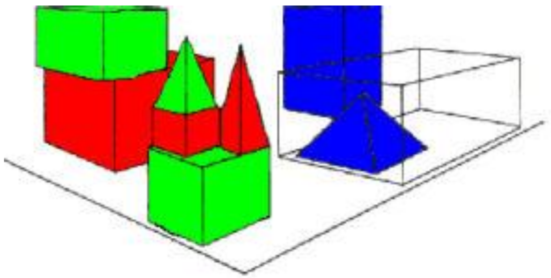


Navigation

Approach: Heavy reliance on domain knowledge

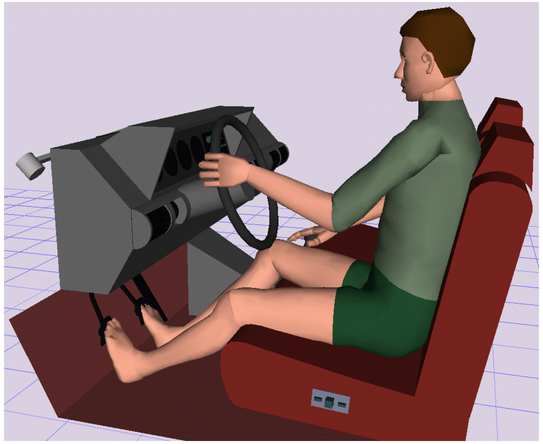
Interpreting High-level Instructions: An Old AI Problem

Winograd, 1972



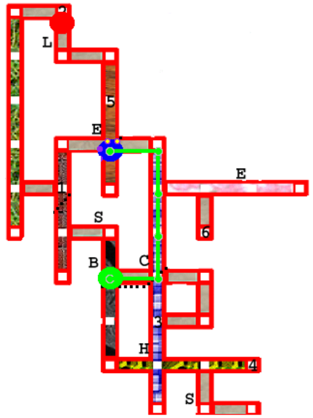
Block-worlds

Di Eugenio, 1992; Webber, 1995



Virtual character animation

MacMahon, 2006



Navigation

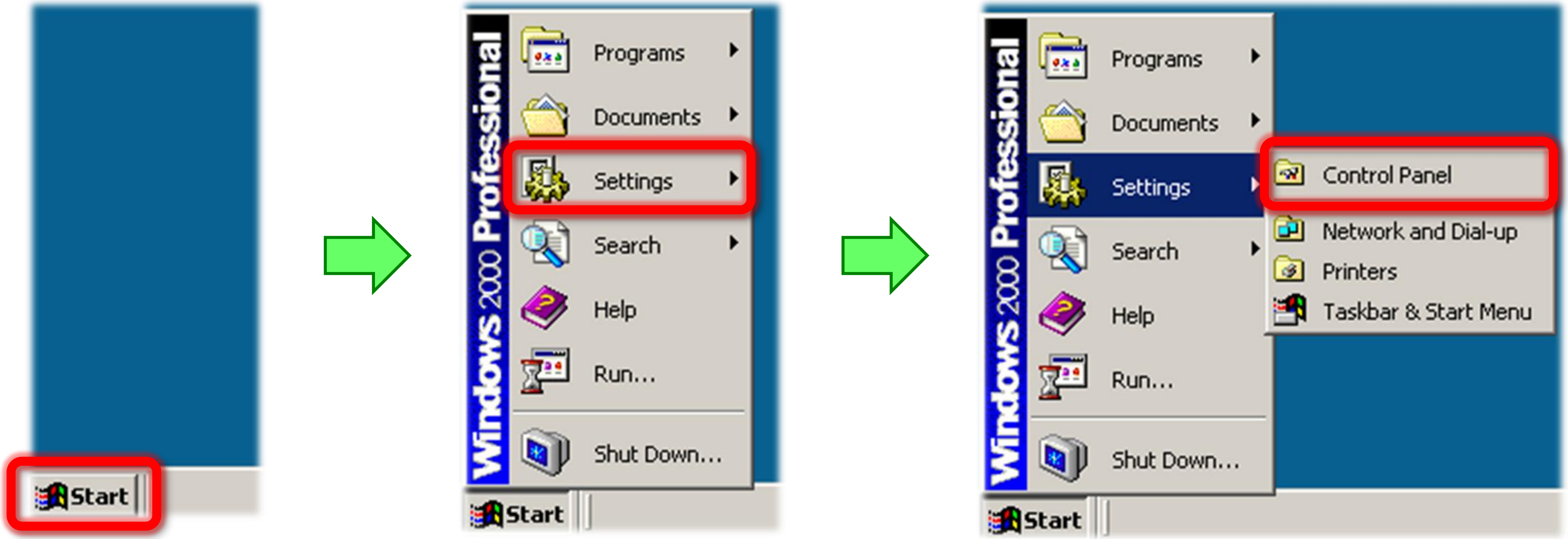
manually specified

Approach: Heavy reliance on domain knowledge

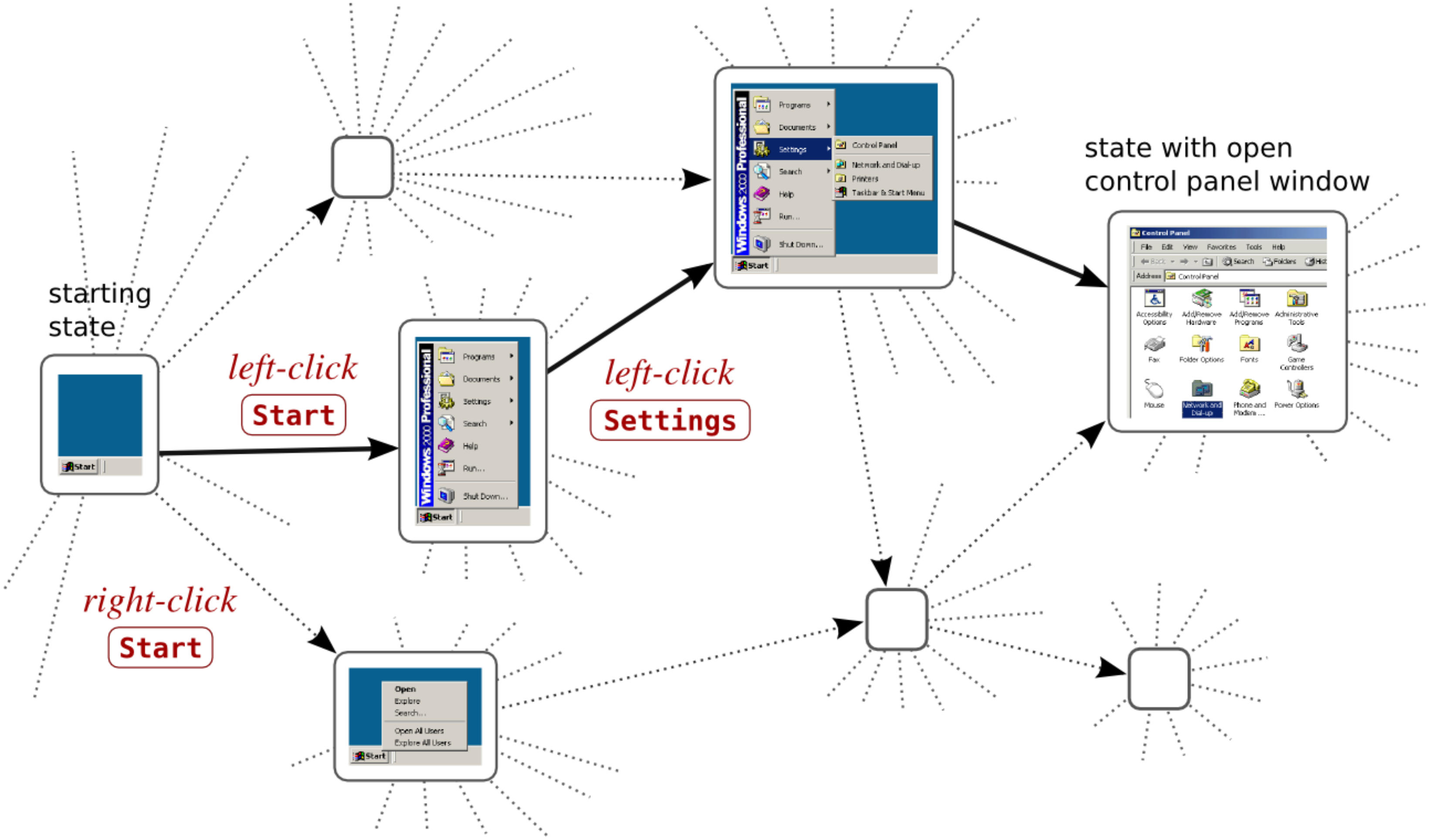
Humans Interpretation of High-level Instructions

High-level instruction: ***“open control panel”***

left-click `Start` *left-click* `Settings` *left-click* `Control Panel`



Environment Model as a State Transition Graph



state with open control panel window

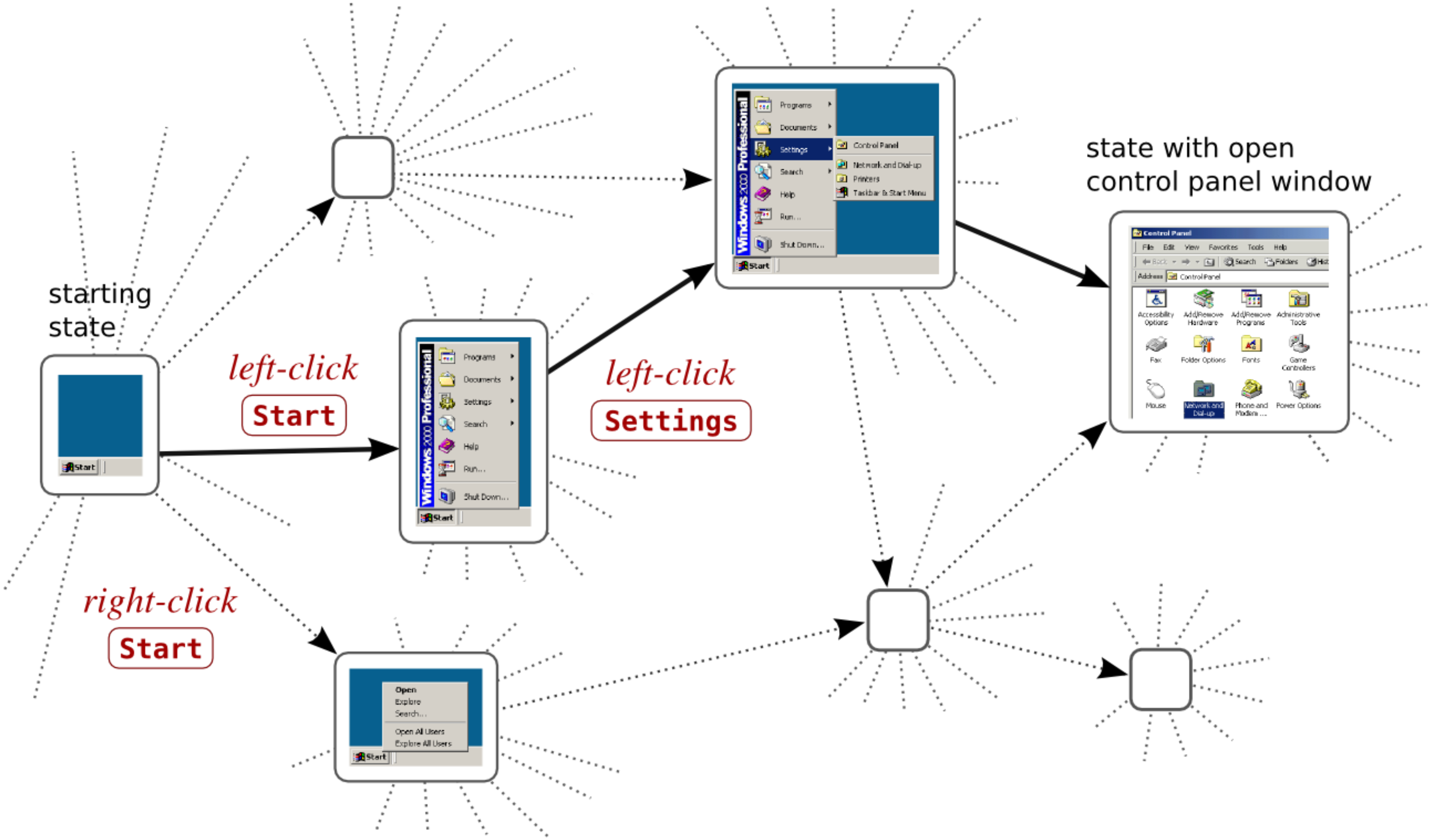
starting state

left-click **Start**

left-click **Settings**

right-click **Start**

Environment Model as a State Transition Graph

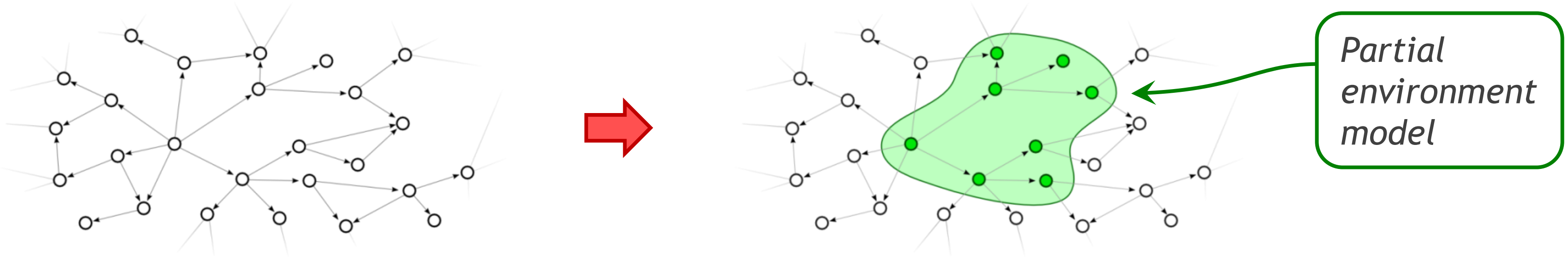


- *Extremely large graph*
- *Need to learn a relevant portion of graph*

Key Idea

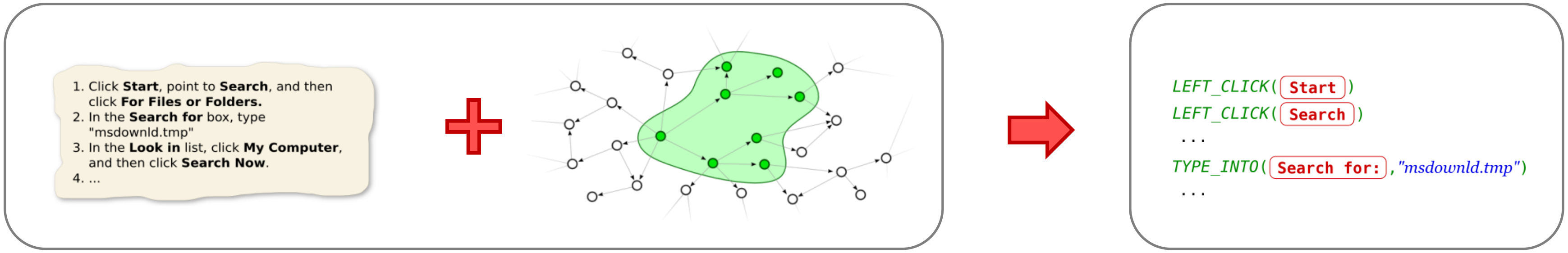
Ground language interpretation in environment dynamics

1. *Automatically construct a relevant environment model*



Partial environment model

2. *Incorporate environment model into language interpretation algorithm*



1. Click **Start**, point to **Search**, and then click **For Files or Folders.**
2. In the **Search for** box, type "msdownld.tmp"
3. In the **Look in** list, click **My Computer**, and then click **Search Now**.
4. ...

```
LEFT_CLICK(Start)
LEFT_CLICK(Search)
 ...
TYPE_INTO(Search for:,"msdownld.tmp")
 ...
```

Reinforcement Learning Framework for Mapping Low-level Instructions

(*ACL 2009*)

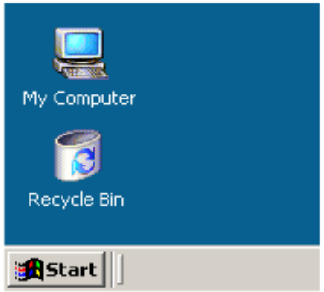
Reinforcement Learning for Instruction Mapping

(ACL 2009)

State s

Observed text and environment

Select run after clicking start. In the open box type "dcomcnfg".



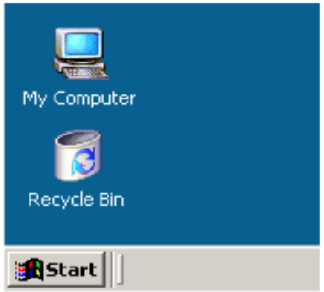
Reinforcement Learning for Instruction Mapping

(ACL 2009)

State s

Observed text and environment

Select run after clicking start. In the open box type "dcomcnfg".



Action a

word span:
clicking start

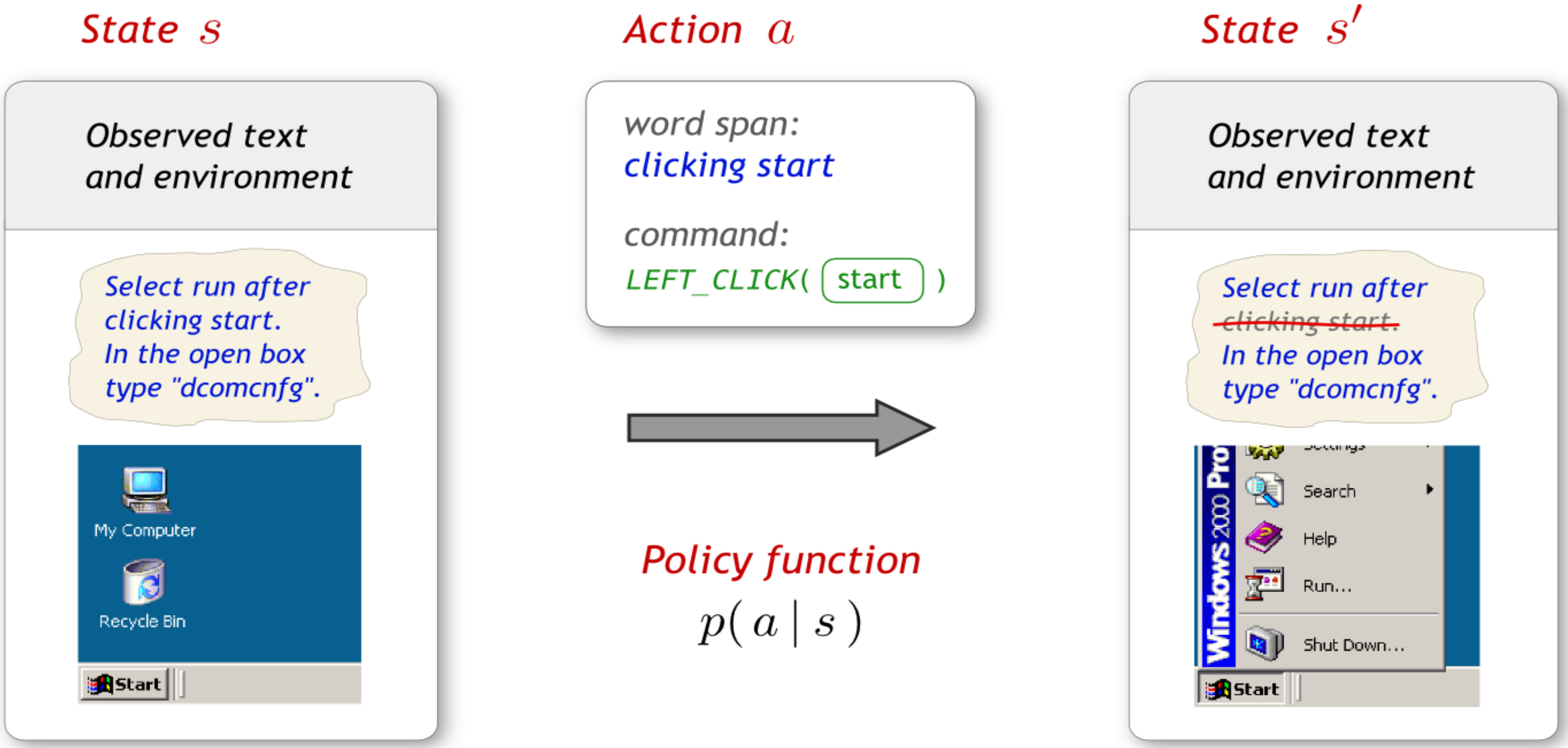
command:
LEFT_CLICK(start *)*

Policy function

$$p(\,a \mid s\,)$$

Reinforcement Learning for Instruction Mapping

(ACL 2009)



State s

Observed text and environment

Select run after clicking start. In the open box type "dcomcnfg".

Action a

word span: *clicking start*

command: *LEFT_CLICK(* start *)*

Policy function

$p(a \mid s)$

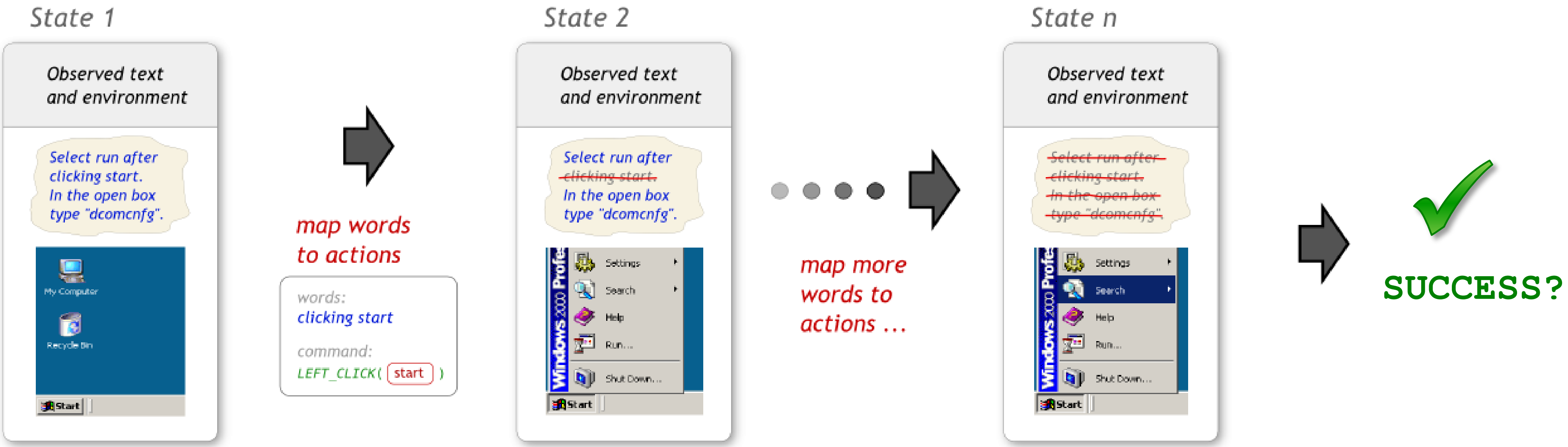
State s'

Observed text and environment

Select run after ~~clicking start.~~ In the open box type "dcomcnfg".

Reinforcement Learning for Instruction Mapping

(ACL 2009)



State 1: Select run after clicking start. In the open box type "dcomcnfg".

words: clicking start

command: LEFT_CLICK(start)

State 2: Select run after ~~clicking start.~~ In the open box type "dcomcnfg".

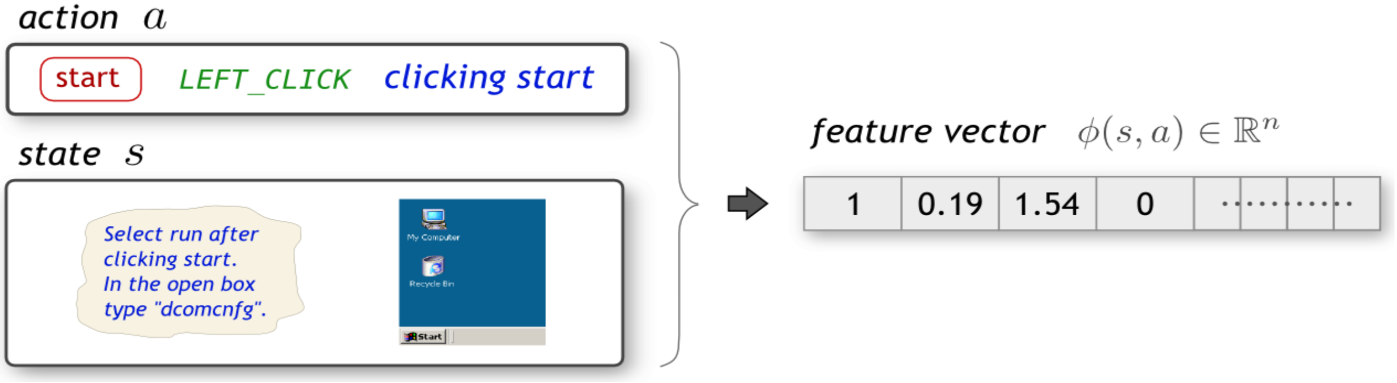
State n: ~~Select run after clicking start. In the open box type "dcomcnfg".~~

- Repeat:
  - *Observe current state of text + environment*
  - *Select action based on policy function*
  - *Execute action*
- Receive reward and update parameters of policy function $p(a|s;\theta)$

Policy Function Parameterization

(ACL 2009)

Represent each action with a feature vector:



action a

start LEFT_CLICK clicking start

state s

Select run after clicking start. In the open box type "dcomcnfg".

feature vector $\phi(s,a) \in \mathbb{R}^n$

1 | 0.19 | 1.54 | 0 | ⋯

e.g., Binary features on [*word*, *command*] pairs

Define policy function as a log-linear distribution:

$$p(a \mid s; \theta) = \frac{e^{\theta \cdot \phi(s,a)}}{\sum_{a'} e^{\theta \cdot \phi(s,a')}}$$

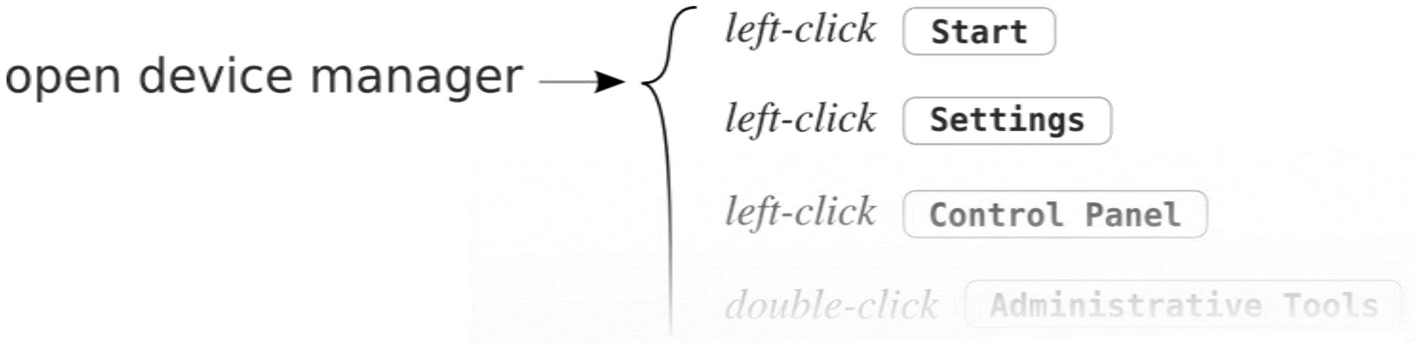
θ - *parameters of model*

$\phi(s,a) \in \mathbb{R}^n$ - *real valued feature function on state s and action a*

Limitations of Framework

Every command is explicitly specified in the text

→ *Cannot handle high-level instructions*



open device manager →

left-click Start

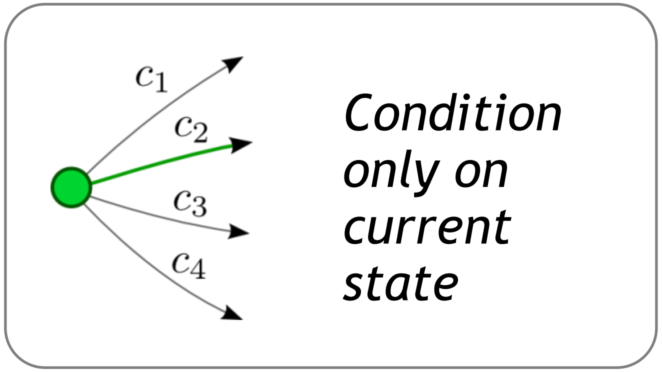
left-click Settings

left-click Control Panel

double-click Administrative Tools

Command selection depends only on current state

→ *Cannot condition on future states*



Condition only on current state

Reinforcement Learning Framework for Mapping High-level Instructions

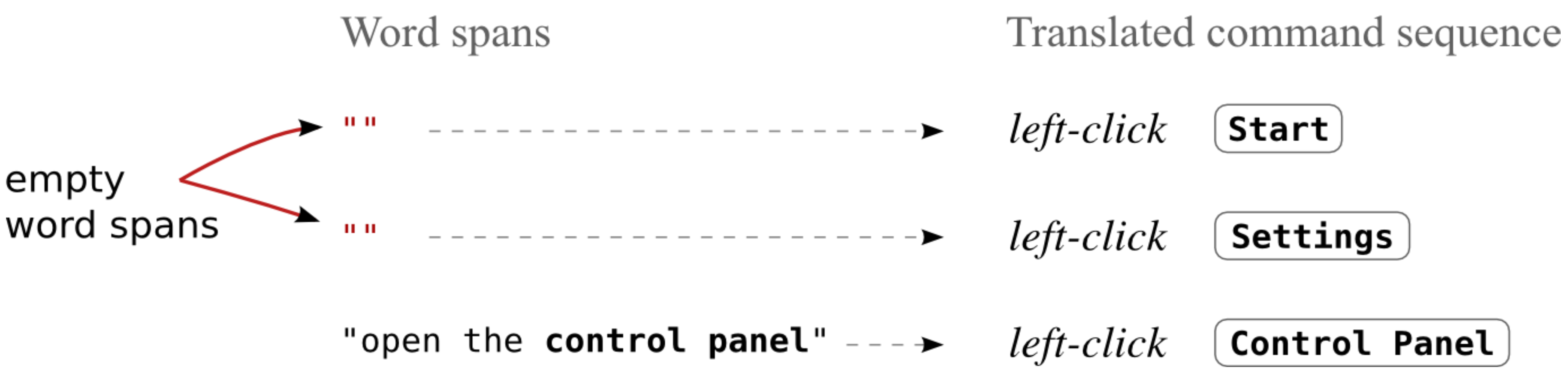
(*current work*)

Highlights of Solution

One-to-many mappings:

1. Allow mapping of empty text to commands

text: "*open the control panel*"



empty word spans

Word spans	Translated command sequence
""	*left-click* Start
""	*left-click* Settings
"open the **control panel**"	*left-click* Control Panel

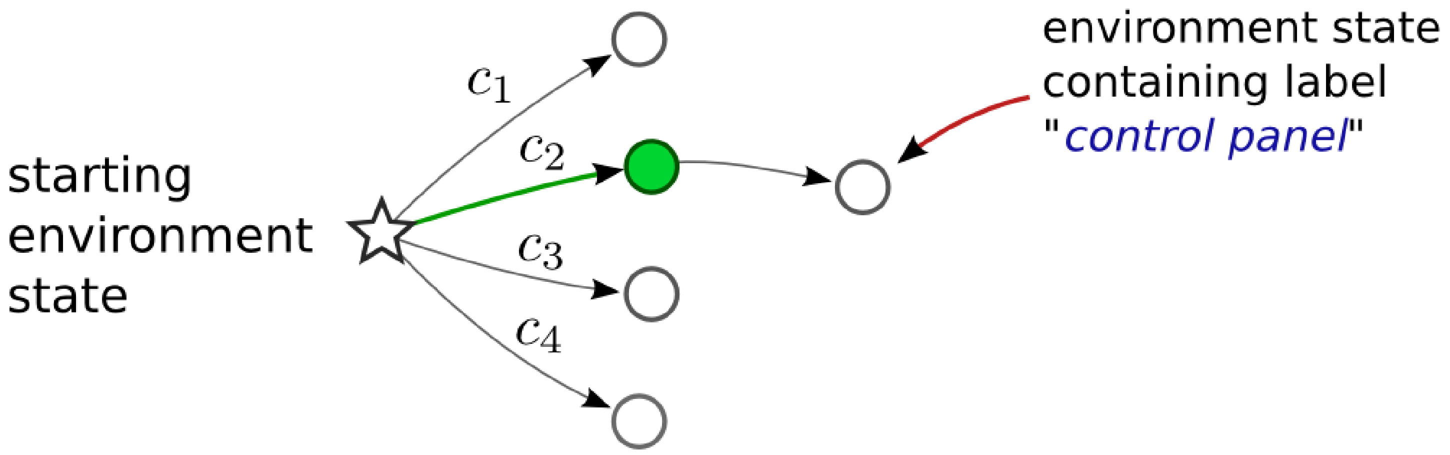
- *Uncertainty increases dramatically*
- *Search space gets much larger*

Highlights of Solution

One-to-many mappings:

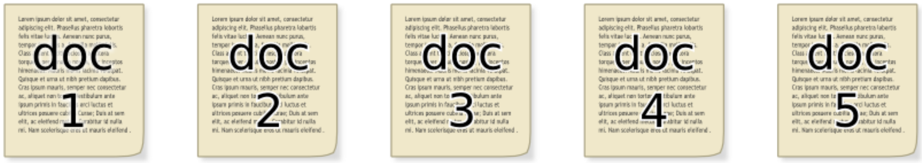
2. Conditioning on future states & unmapped text

text: "*open the control panel*"

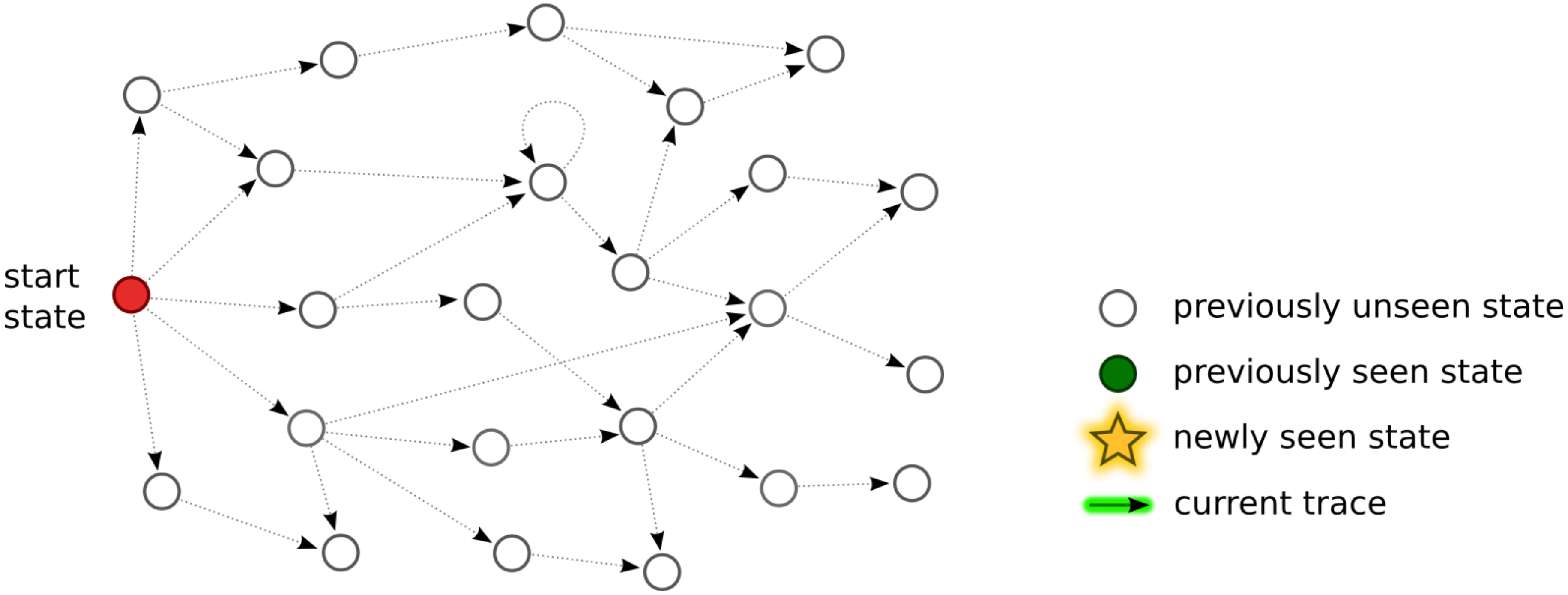

starting environment state
c_1
c_2
c_3
c_4
environment state containing label "*control panel*"

- *Cannot learn a complete environment model, Need to learn a relevant partial model*

Acquiring a Partial Environment Model



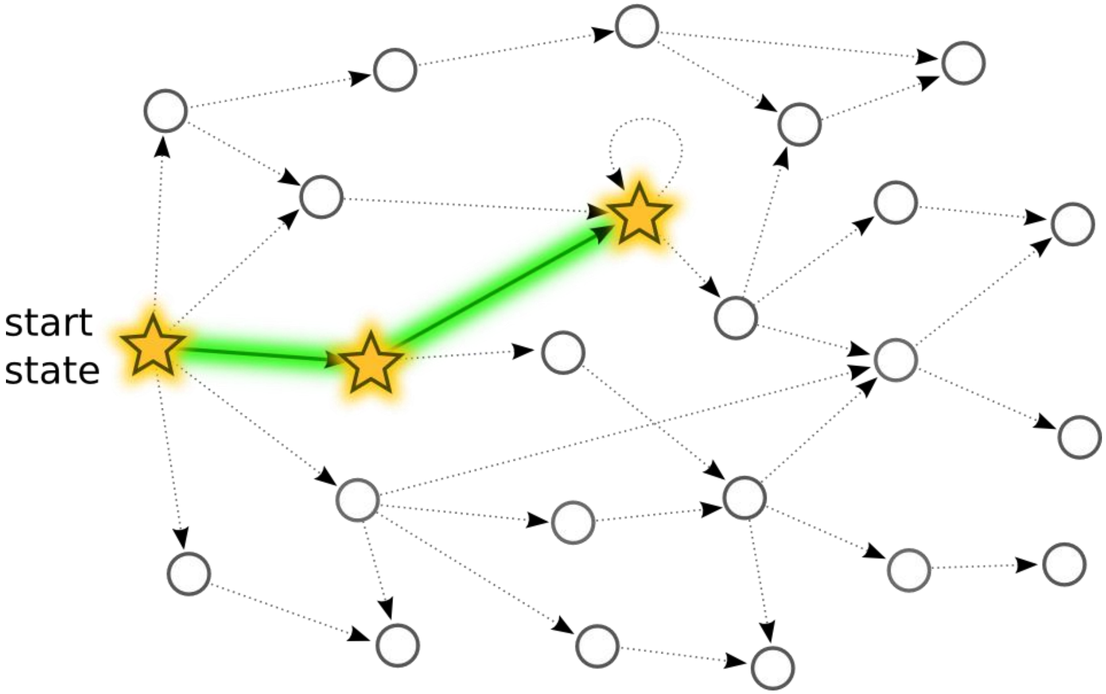
Input documents



Acquiring a Partial Environment Model

doc 1
doc 2
doc 3
doc 4
doc 5

Interpretation 1



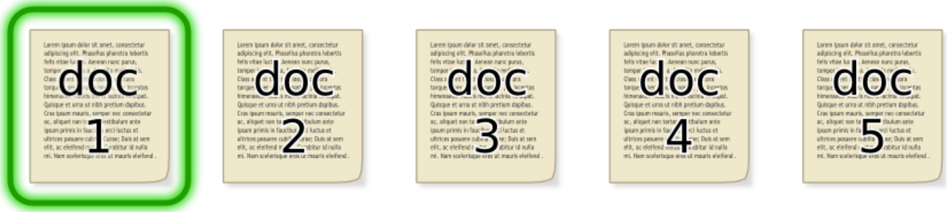
○ previously unseen state

● previously seen state

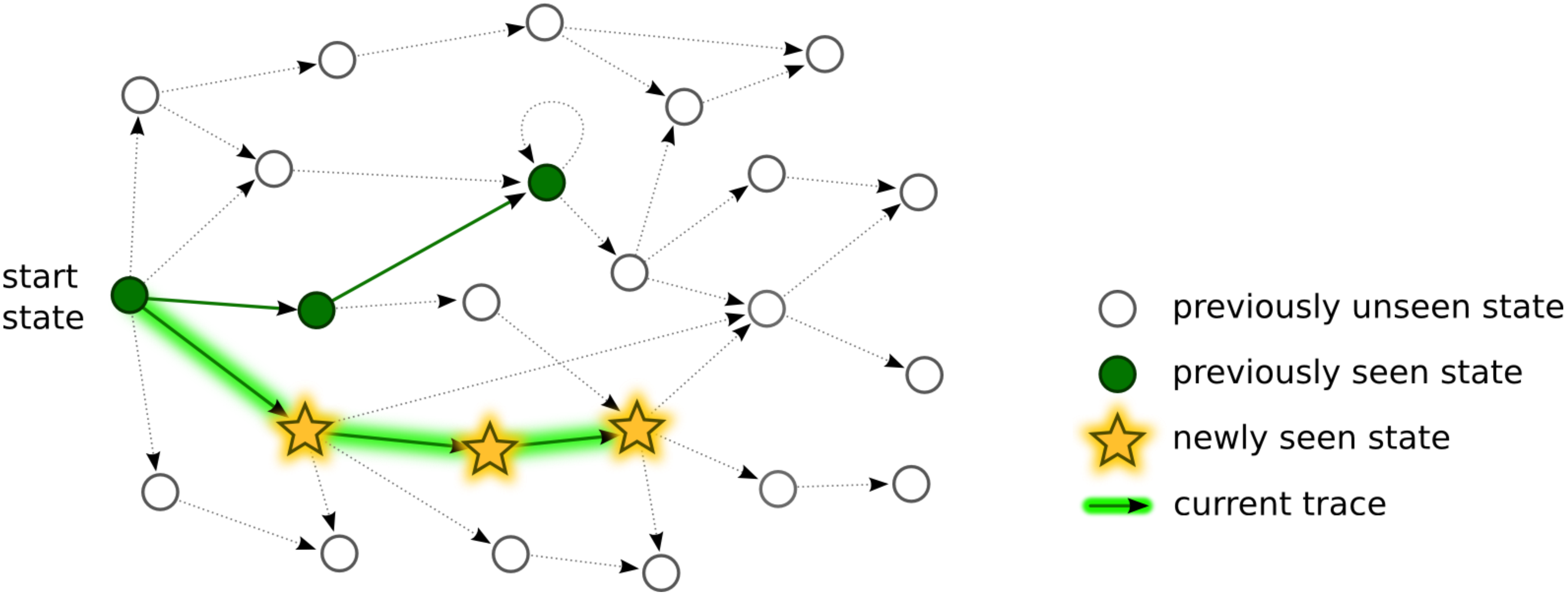
☆ newly seen state

→ current trace

Acquiring a Partial Environment Model



Interpretation 2


start state

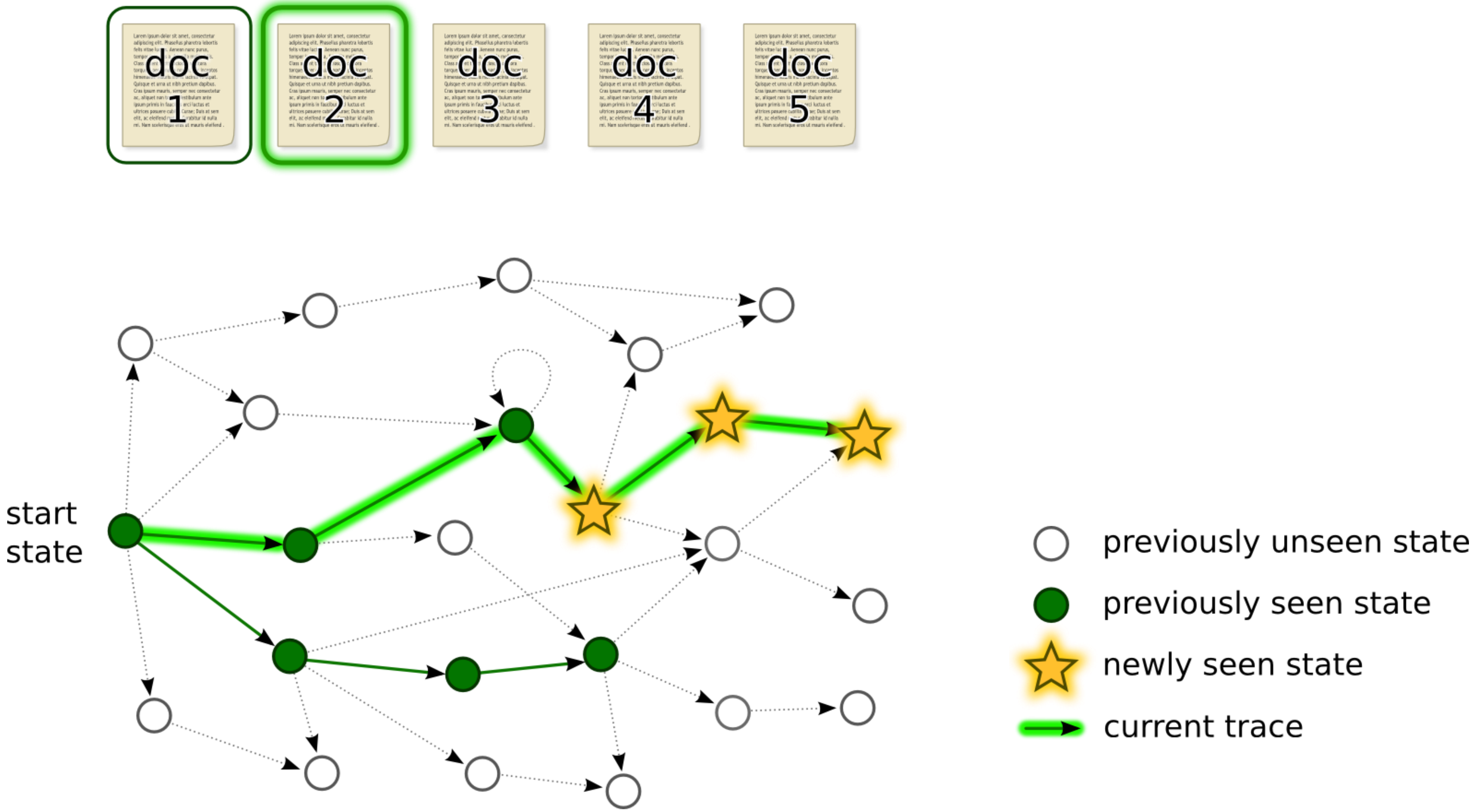
○ previously unseen state

● previously seen state

☆ newly seen state

→ current trace

Acquiring a Partial Environment Model



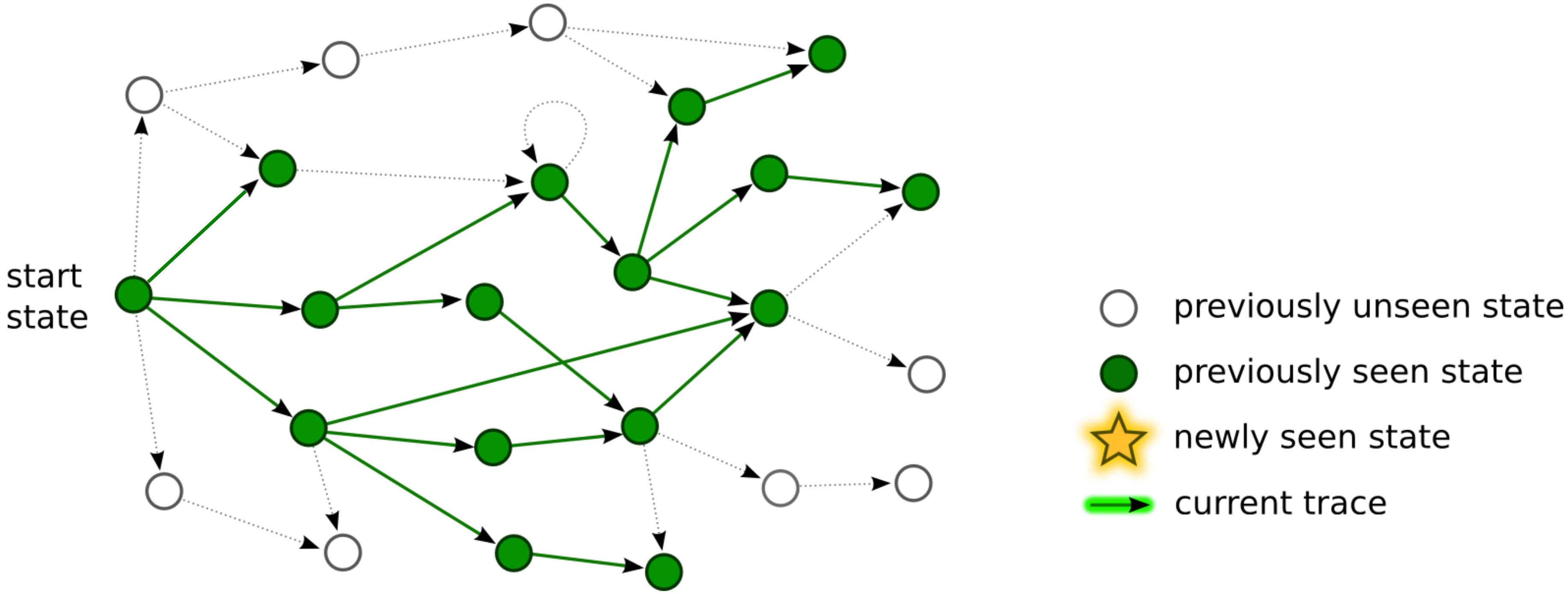
○ previously unseen state

● previously seen state

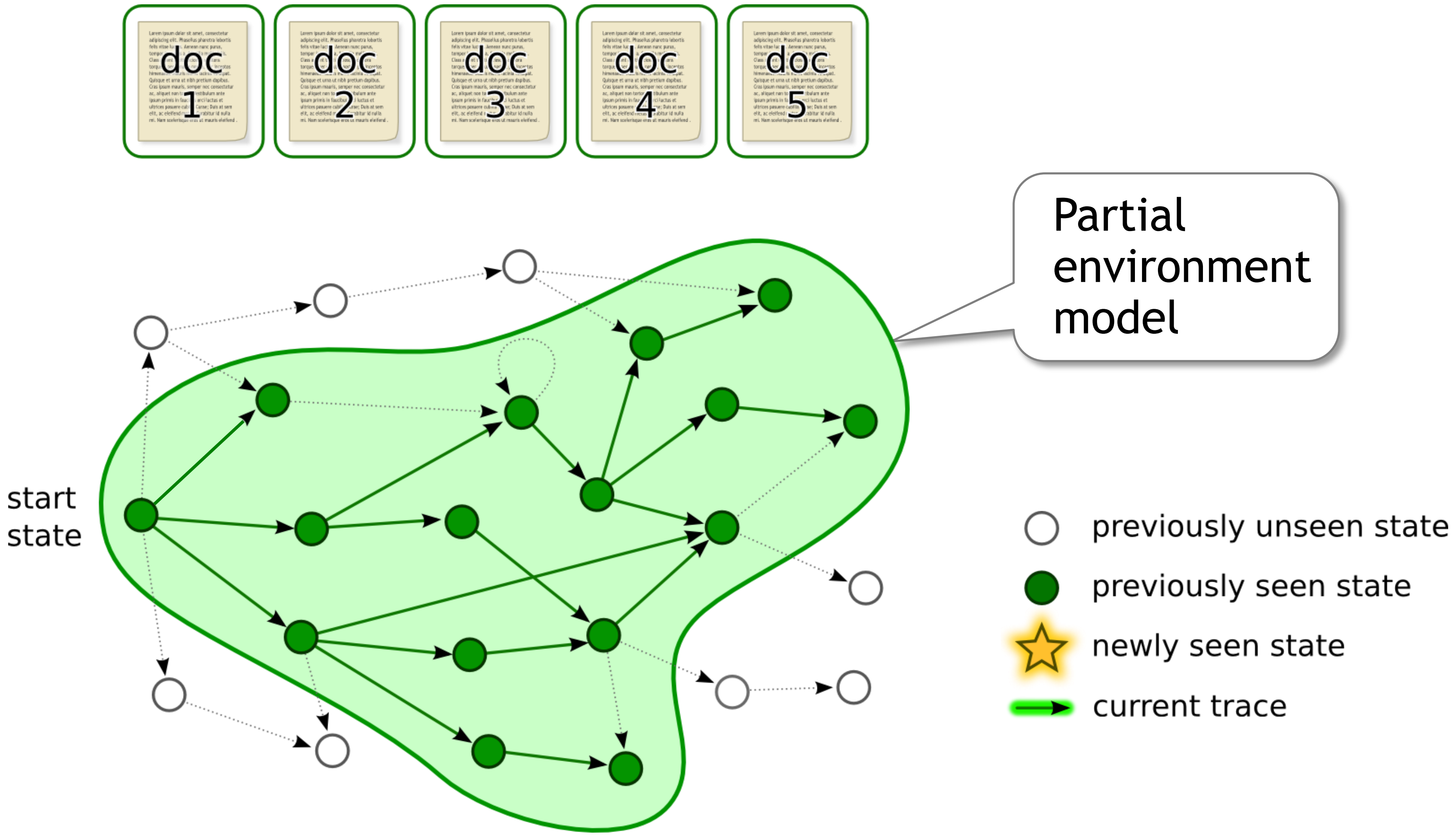
☆ newly seen state

→ current trace

Acquiring a Partial Environment Model



Acquiring a Partial Environment Model



Partial environment model

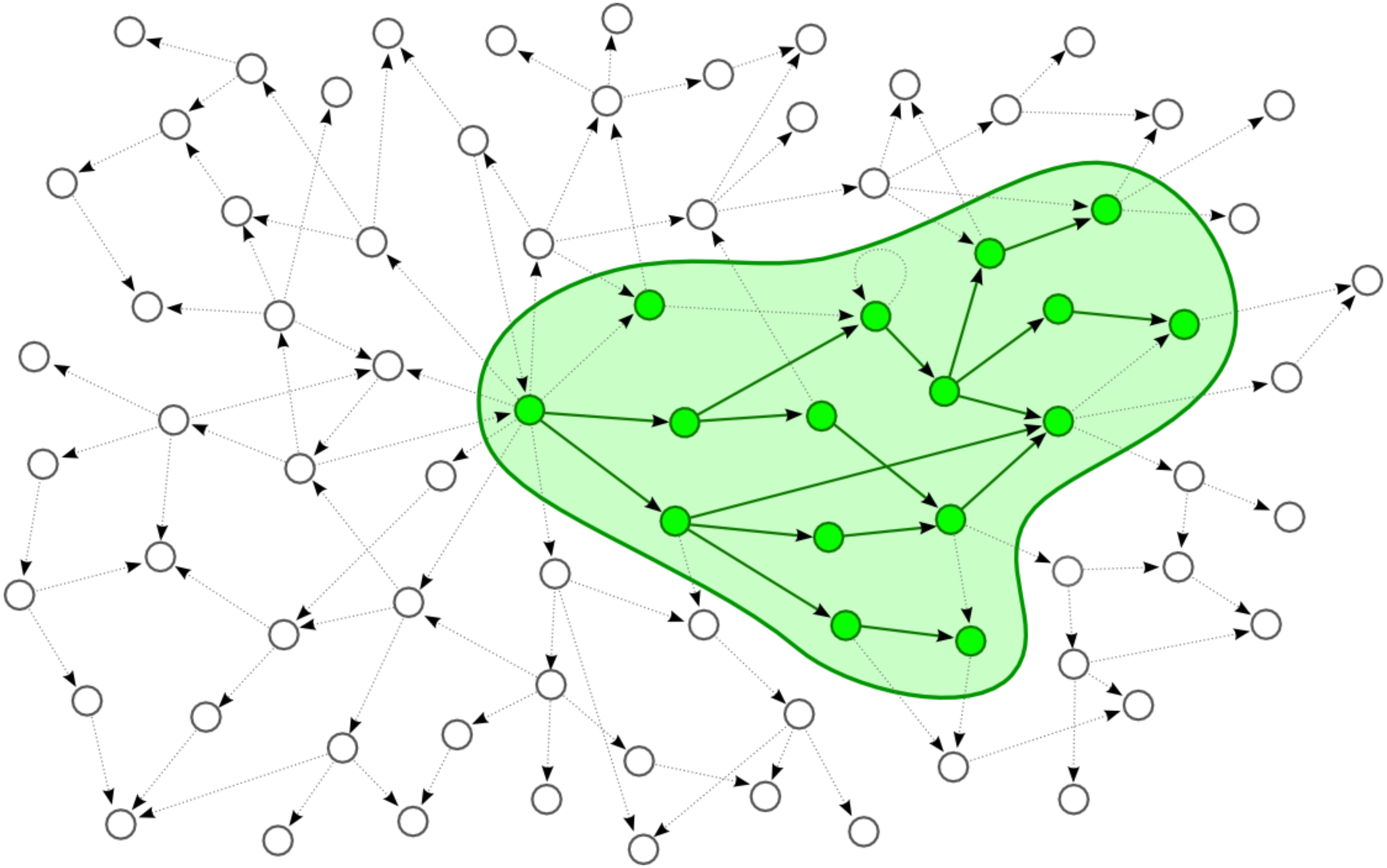
○ previously unseen state

● previously seen state

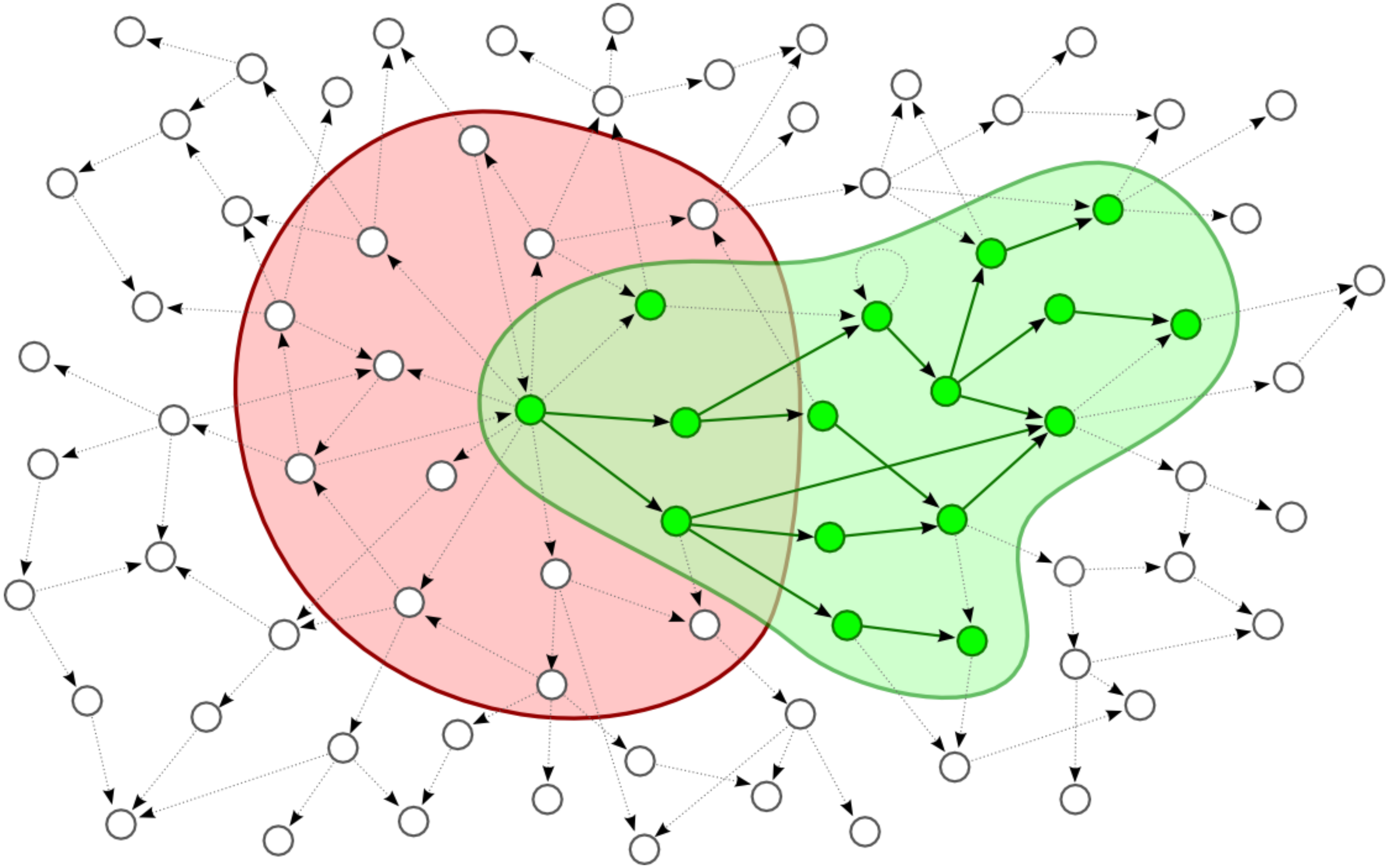
☆ newly seen state

→ current trace

Acquiring a Partial Environment Model



Acquiring a Partial Environment Model



Exploiting The Partial Environment Model

Select commands using information about future states

Encode future state attributes as features, e.g.:

- *Average reward*
- *Length of path to potential goals*

$\phi(a, s)$ ⇨ $\phi(a, s, q)$

$p(a \mid s; \theta)$ ⇨ $p(a \mid s; q, \theta)$

q - *partial environment model*

Policy Function

Policy for selecting actions:

$$p(a \mid s; q, \theta) = p(\vec{w} \mid s; \theta_w) \quad \times \quad p(c \mid \vec{w}, s; q, \theta_c)$$

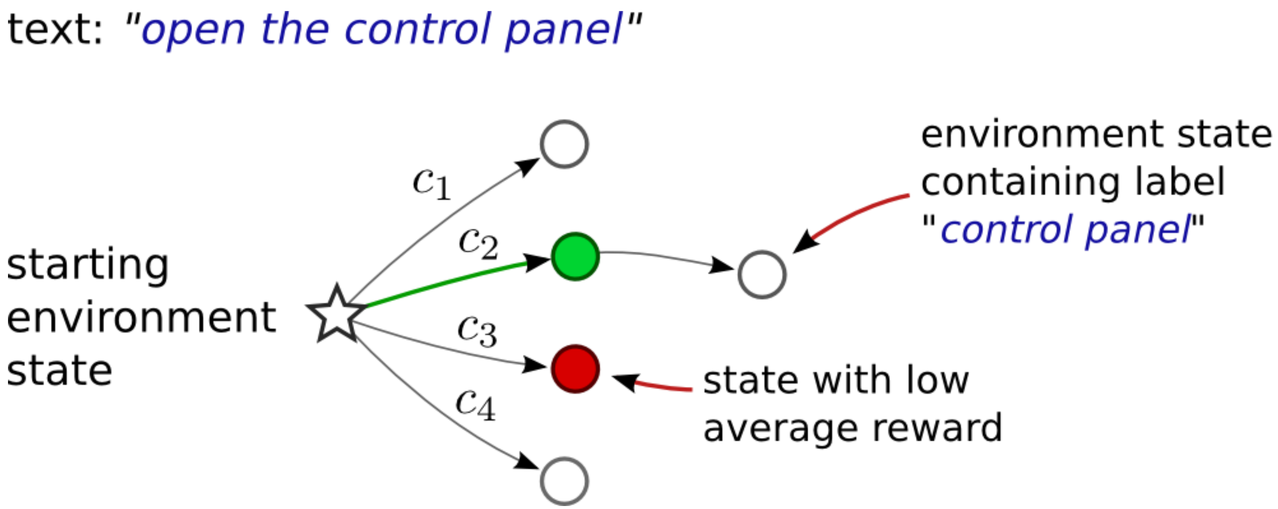
s	- state
a	- action
θ	- model parameters
c	- environment command
$\vec{w}$	- Instruction word span
q	- partial environment model

Features from Environment Model

1. Highest reward achievable from current state by any action sequence

2. Length of highest-reward action sequence

3. Average reward received at environment state for any document



text: "*open the control panel*"

starting environment state

c_1

c_2

c_3

c_4

environment state containing label "*control panel*"

state with low average reward

Example Local Features

Features on words and environment command & object

- Binary feature on each (*word*, *command*) pair
- Binary feature on each (*word*, *object type*) pair

Features on environment objects

- Object is in foreground
- Object was previously interacted with
- Object became visible after last action

Features on words

- Word type
- Distance from last used word

Total number of features: 4438

Windows Configuration Application

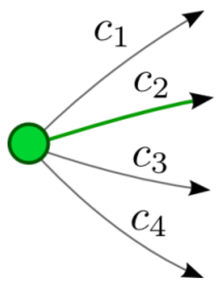
Windows 2000 help documents
from *support.microsoft.com*

Total # of documents	188
# documents with high-level instructions	60
Total # of words	7448
Vocabulary size	739
Avg. commands per document	10

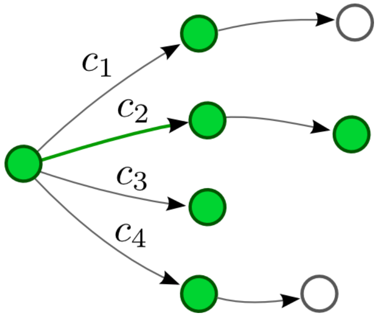
Evaluation Metric: Command prediction accuracy
on heldout documents

Evaluation

1. How important is the environment model for interpreting high-level instruction?



Condition only on current state

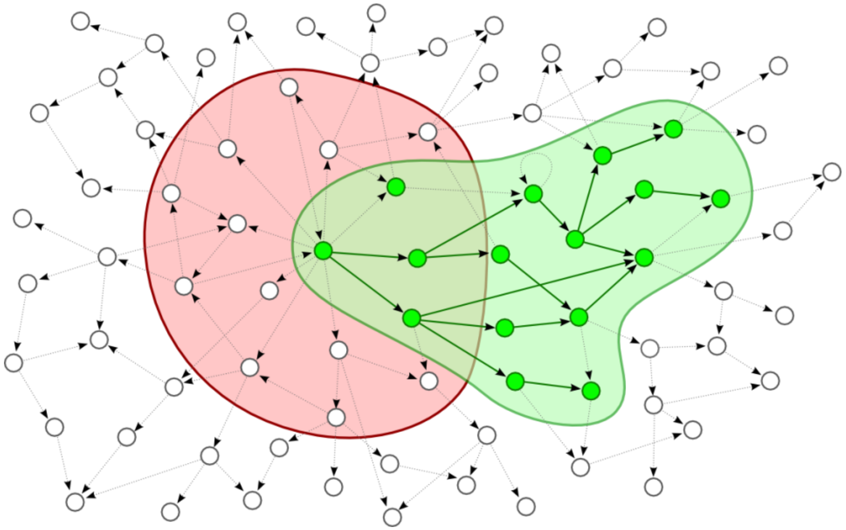


Condition on current and future states

Evaluate against method without environment model

Evaluation

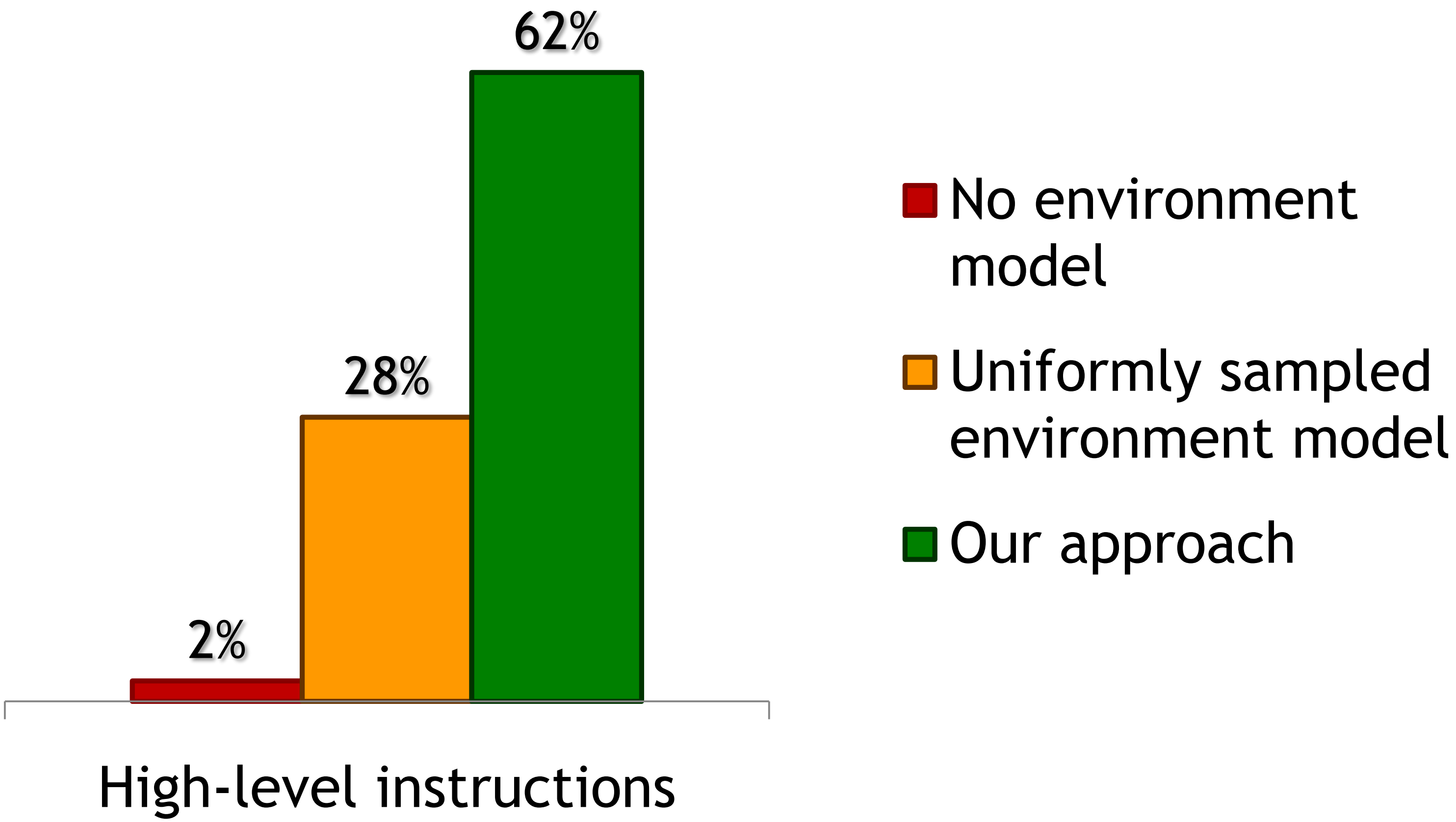
2. How useful is document information in constructing the environment model?



Compare against method with uniformly sampled environment model

Results: High-level Instructions

Command accuracy



Automatic Paraphrasing of High-level Instructions

High-level instruction

- open **device manager**

↓

Low-level instruction paraphrase

- double click **my computer**
- double click **control panel**
- double click **administrative tools**
- double click **computer management**
- double click **device manager**

Conclusions

1. *Domain knowledge is essential for high-level instructions*
2. *Relevant domain knowledge can be acquired automatically*

Software environments

Other domains?

Code and data available at:

groups.csail.mit.edu/rbg/code/