GRAPH-BASED SEMANTIC PARSING, COMPOSITIONAL GENERALIZATION AND LOSS FUNCTIONS

Caio Corro
Université Paris-Saclay, LISN, CNRS
https://caio-corro.fr

SEMANTIC PARSING

Related publication

On graph-based reentrancy-free semantic parsing

Alban Petit, Caio Corro

TACL 2023

SEMANTIC PARSING

SQL parsing

- Input: sentence
- Output: SQL query

```
I want to book a flight from Paris to Rome.
```

$\Downarrow$

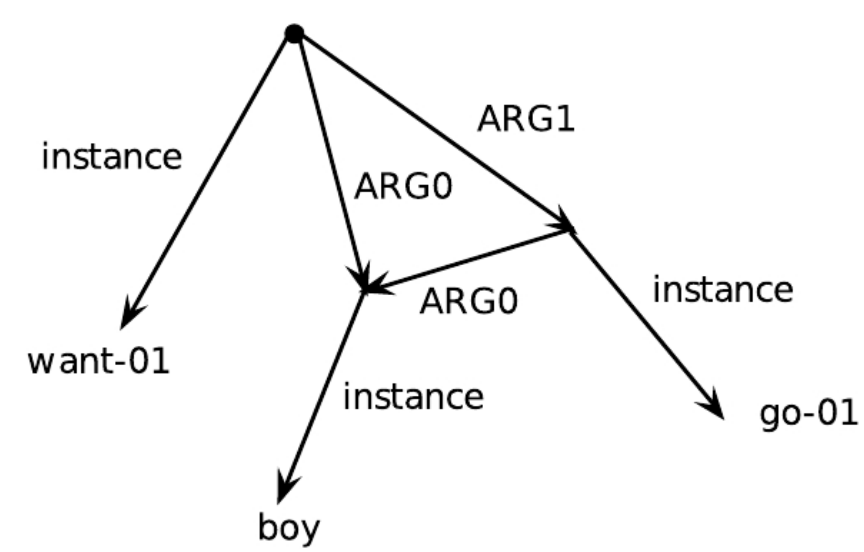
```
SELECT * FROM  flight WHERE  from = "paris" AND to = "rome"
```

Abstract Meaning Representation (AMR) parsing

- Input: sentence
- Output: graph

```
The boy want to go.
```

$\Rightarrow$



REENTRANCY-FREE SEMANTIC PARSING

Reentrancy-free semantic structures

- Predicates and entities are typed (in the same sense than in “typed programming languages")
- An argument can only be used once

Semantic structures look like a simple instruction in a functional programming language.

```
What   rivers   do   not   run  through  Tennesse?
```

$$\Downarrow$$

```
exclude ( river_all , traverse_2 ( stateid('Tennesse') ) )
```

Is this realistic?

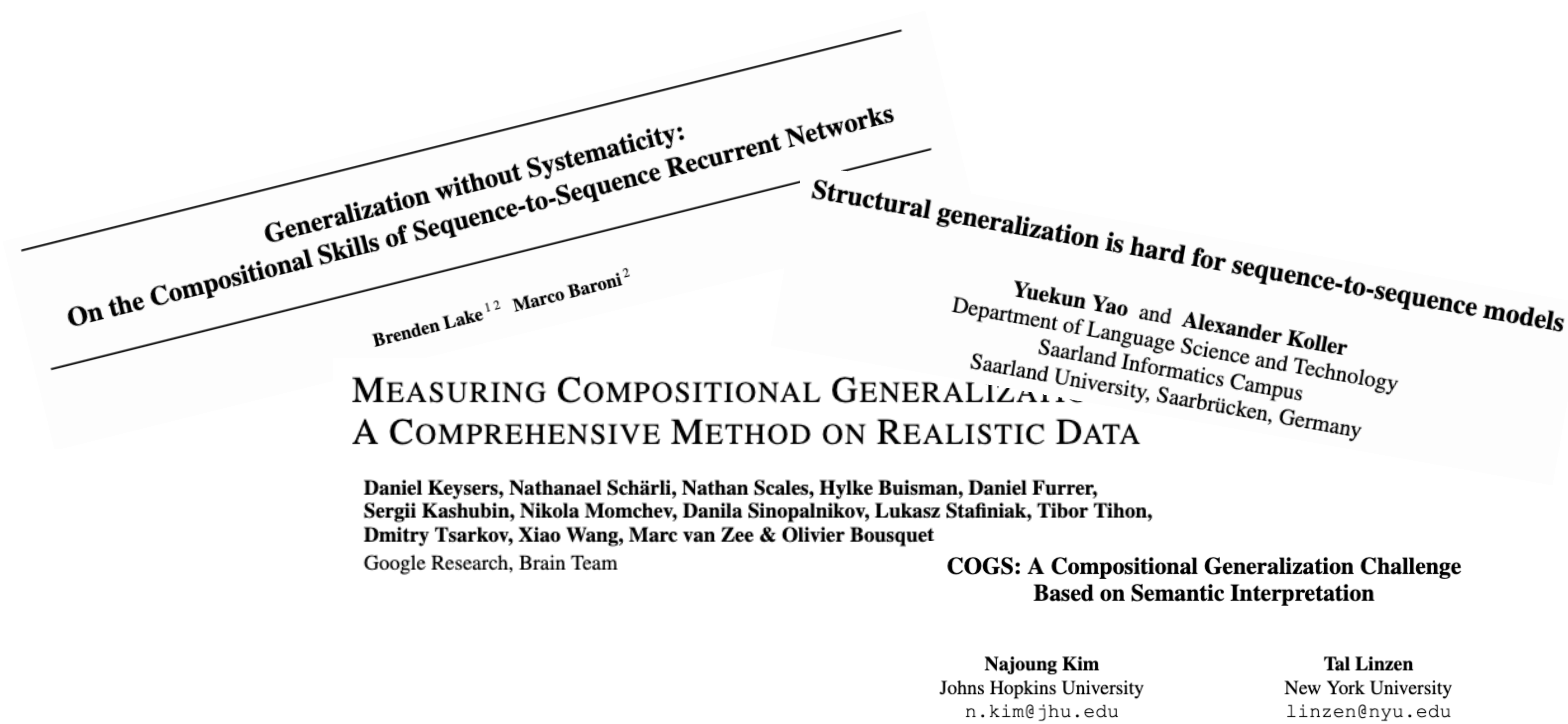
"estimating that there are only 0.3% queries that would require a more general [..] representation."

Task Oriented Parsing (TOP) dataset **[Gupta et al., 2018]**

COMPOSITIONAL GENERALIZATION

Compositionality: "the meaning of a complex expression is constructed from the meanings of its constituent parts" (Kim & Linzen, 2020)

Compositional generalization: "Once a person learns the meaning of a new verb *dax*, he or she can immediately understand the meaning of *dax twice* or *sing and dax*." (Lake & Baroni, 2018)



Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks

Brenden Lake[1 2] Marco Baroni[2]

Structural generalization is hard for sequence-to-sequence models

Yuekun Yao and **Alexander Koller**
Department of Language Science and Technology
Saarland Informatics Campus
Saarland University, Saarbrücken, Germany

MEASURING COMPOSITIONAL GENERALIZA…
A COMPREHENSIVE METHOD ON REALISTIC DATA

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee & Olivier Bousquet
Google Research, Brain Team

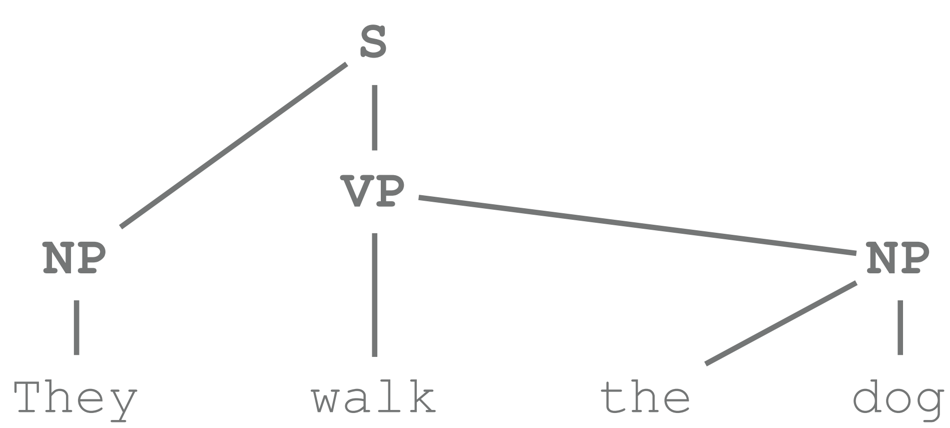
COGS: A Compositional Generalization Challenge Based on Semantic Interpretation

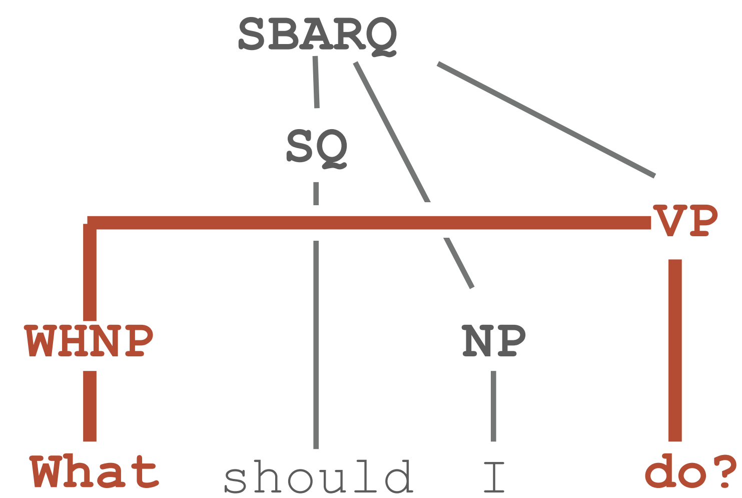
Najoung Kim
Johns Hopkins University
n.kim@jhu.edu

Tal Linzen
New York University
linzen@nyu.edu

GRAPH-BASED SEMANTIC PARSING

SYNTACTIC PARSING: CONSTITUENCY PARSING



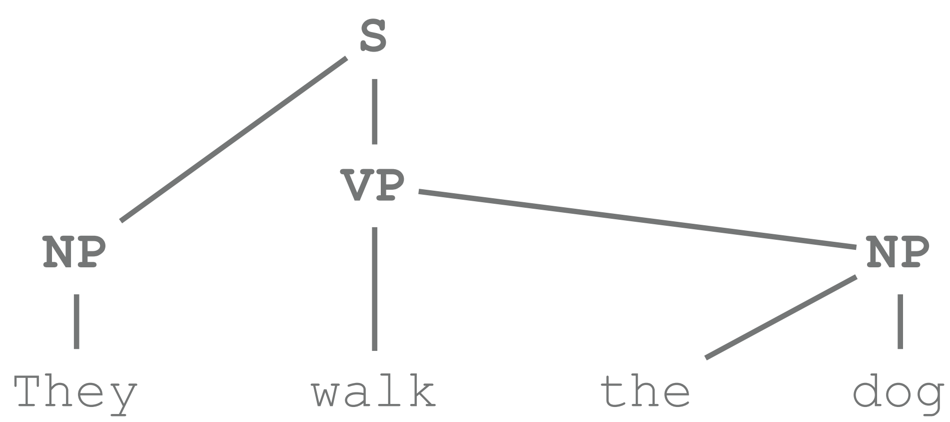


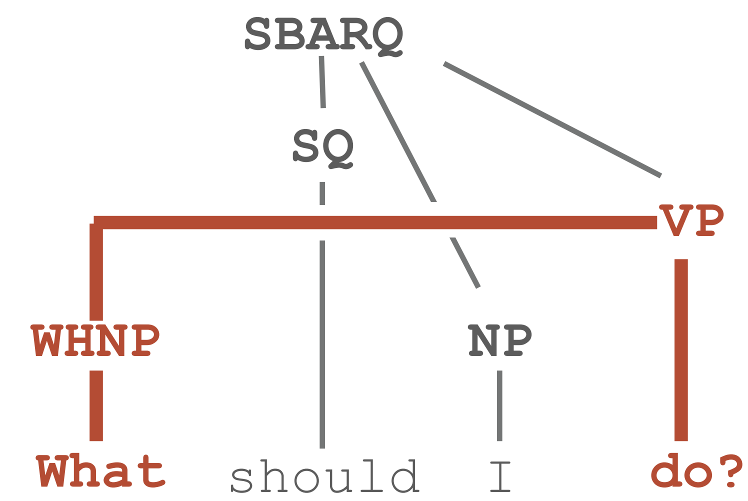
Constituency parsing complexity with formal grammars

Increasing search space

Context-free grammars	$\mathcal{O}(n^3)$	[Sakai, 1961]
Well-nested LCFRS with a fan-out of 2	$\mathcal{O}(n^6)$	[Gómez-Rodríguez et al., 2010]
Well-nested LCFRS with a fan-out of k, k > 2	$\mathcal{O}(n^{2k+2})$	
LCFRS with bounded fan-out	NP-hard	[Satta, 1992]

SYNTACTIC PARSING: CONSTITUENCY PARSING





Constituency parsing complexity with formal grammars

Increasing search space

Context-free grammars	$\mathcal{O}(n^3)$	[Sakai, 1961]
Well-nested LCFRS with a fan-out of 2	$\mathcal{O}(n^6)$	[Gómez-Rodríguez et al., 2010]
Well-nested LCFRS with a fan-out of k, k > 2	$\mathcal{O}(n^{2k+2})$	[Gómez-Rodríguez et al., 2010]
LCFRS with bounded fan-out	NP-hard	[Satta, 1992]

Constituency parsing complexity with span-based parsers

- Ensure the well-formedness of the resulting structure
- Do not enforce compliance of the syntactic content represented by the structure (e.g. a verbal phrase is not constrained to contain a verb)

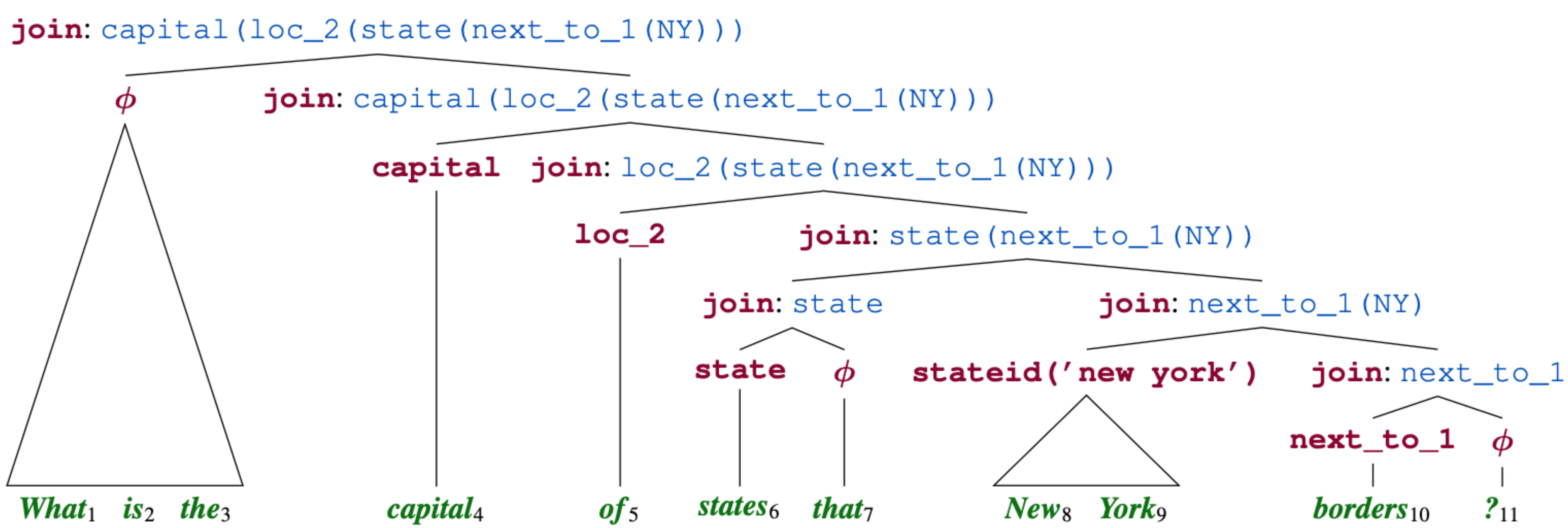
Similar complexity than formal grammar parsers [Stern et al., 2017] [Corro, 2020]

SPAN-BASED SEMANTIC PARSING

[Herzig & Berant, 2021]

Outline

- Use a span-based constituency parser for semantic parsing (with extra valency constraints)
- Show that it is more robust to compositional generalization than seq-2-seq models

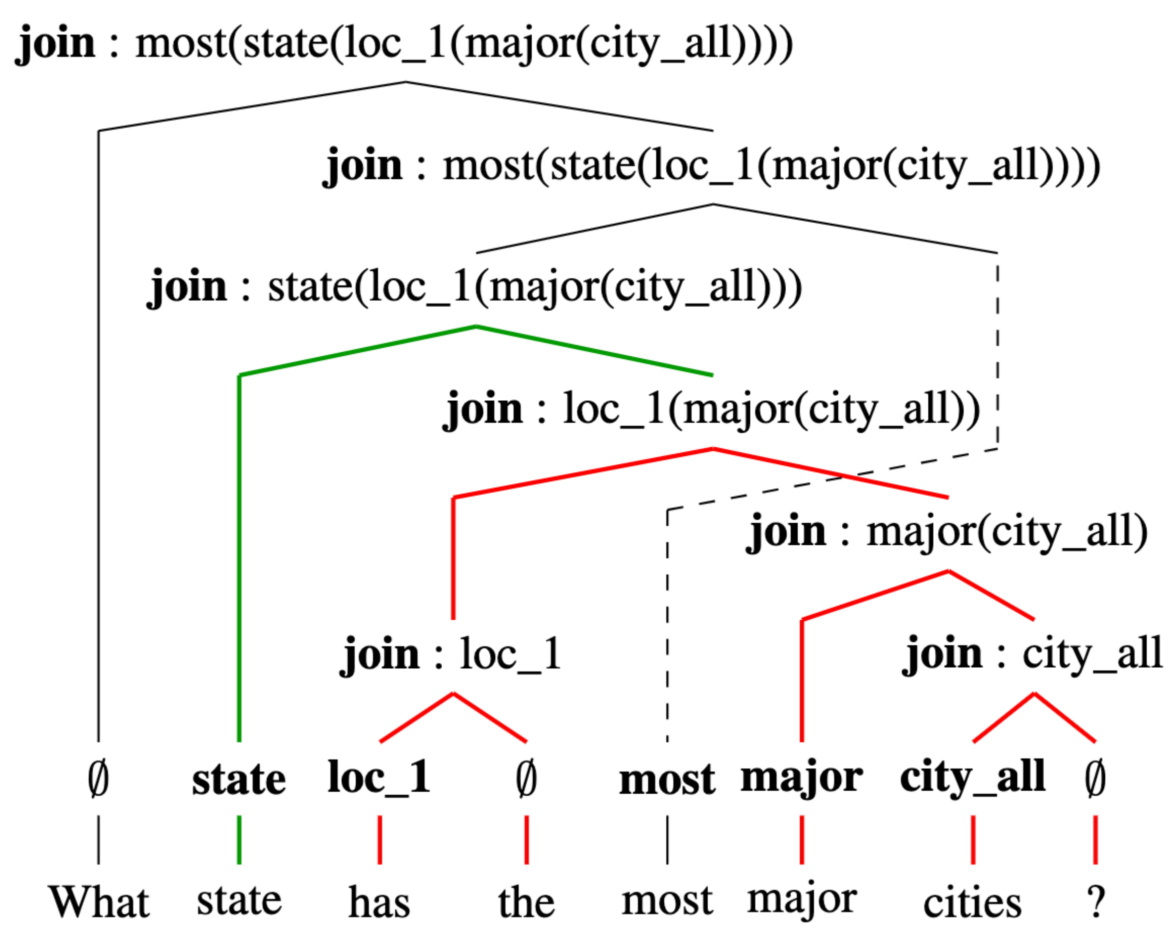


SPAN-BASED SEMANTIC PARSING

[Herzig & Berant, 2021]

Limitation

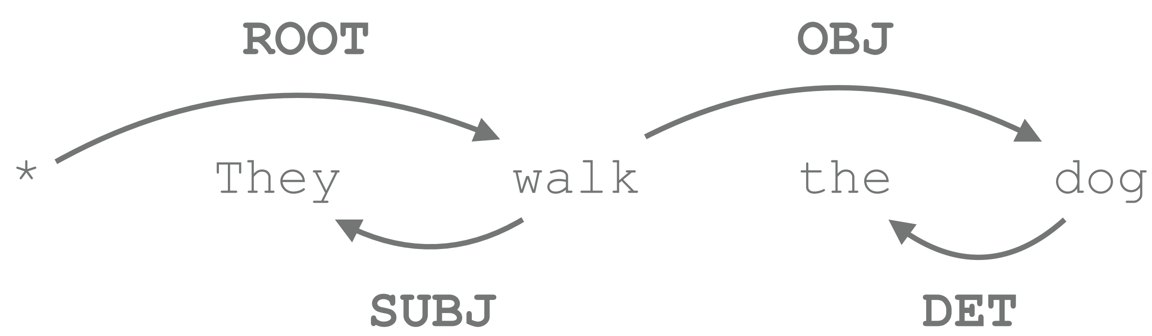
The parser allows only a limited form of discontinuity that can be parsed in $\mathcal{O}(n^3)$ [Corro, 2020]



The constituent in red is discontinuous and also has a discontinuous parent (red+green)

=> outside the search space of the algorithm!

SYNTACTIC PARSING: DEPENDENCY PARSING



Dependency parsing complexity (among many other algorithms!)

Increasing search space

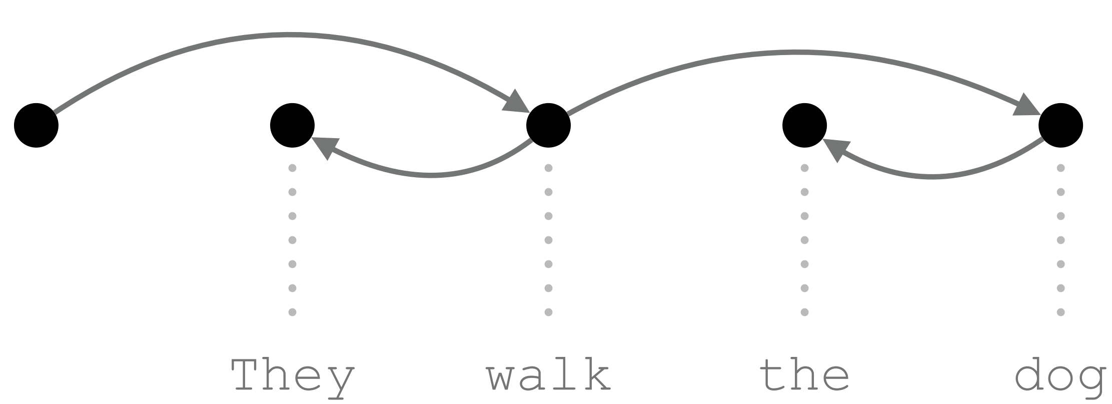
Projective	$\mathcal{O}(n^3)$	[Eisner, 2000]
Well-nested + 2-bounded block degree	$\mathcal{O}(n^7)$	[Gómez-Rodríguez et al. 2009]
Well-nested + k-bounded block degree, k > 2	$\mathcal{O}(n^{3+2k})$	
k-bounded block degree, k > 2	NP-complete	[Satta, 1992]
Unrestricted (a.k.a. non-projective)	$\mathcal{O}(n^2)$	[McDonald et al., 2005] [Tarjan, 1977]

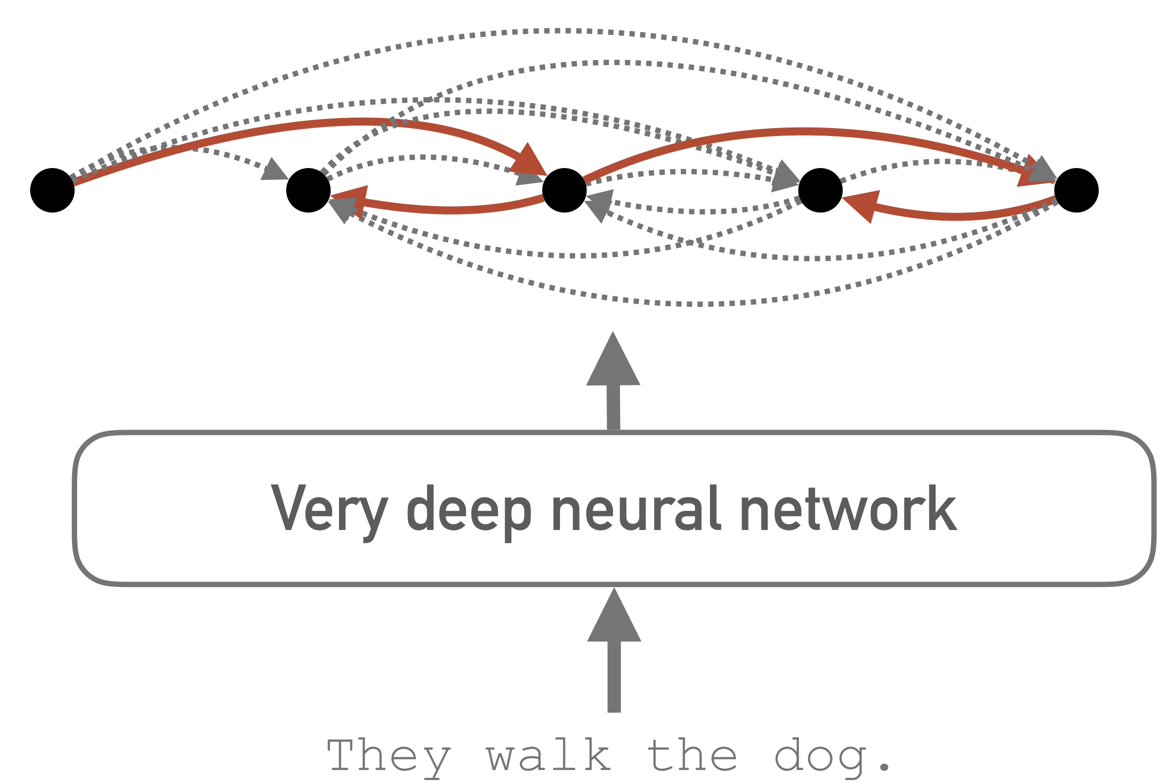
GRAPH-BASED PARSING

Prediction with a graph-based parser

Assume an input sentence with n words:

1. Create a complete directed graph with n vertices
2. Weight all arcs using a neural network
3. Compute the maximum spanning arborescence of the graph





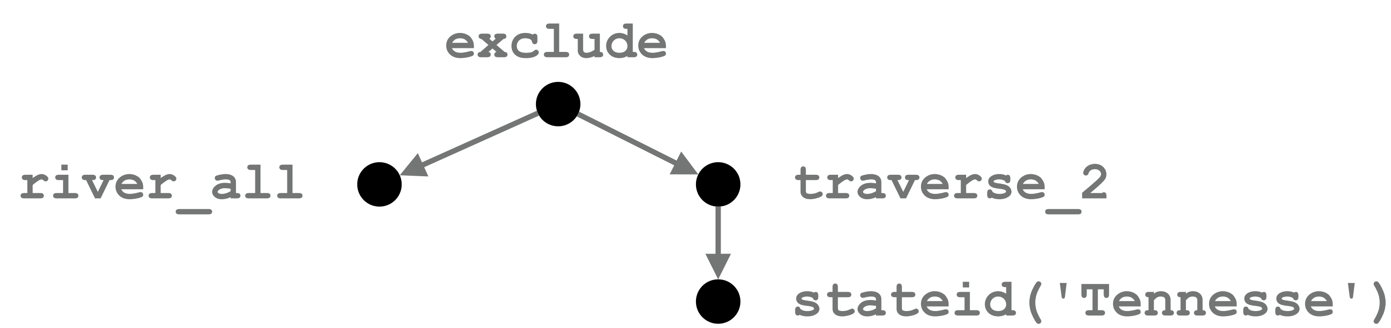
GRAPH-BASED SEMANTIC PARSING

Intuition

The semantic program can be represented by its abstract syntax tree (AST)

=> just predict the AST!

```
exclude ( river_all , traverse_2 ( stateid('Tennesse') ) )
```



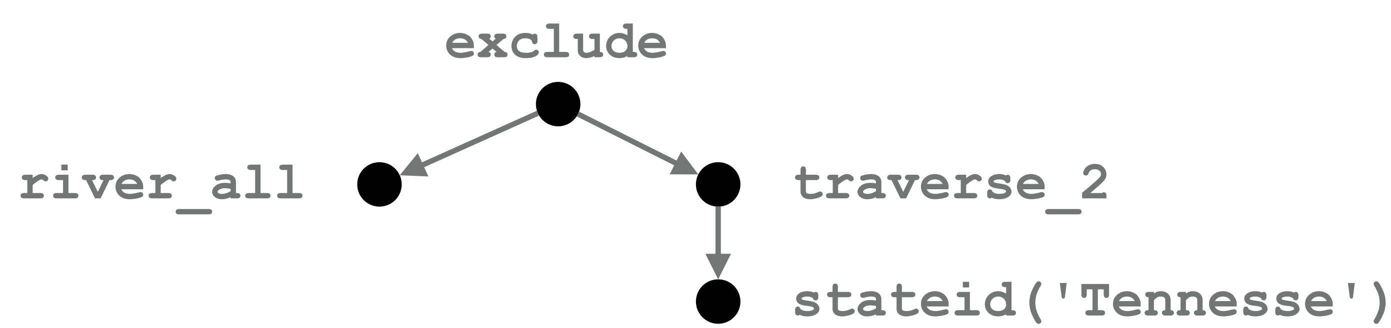
GRAPH-BASED SEMANTIC PARSING

Intuition

The semantic program can be represented by its abstract syntax tree (AST)

=> just predict the AST!

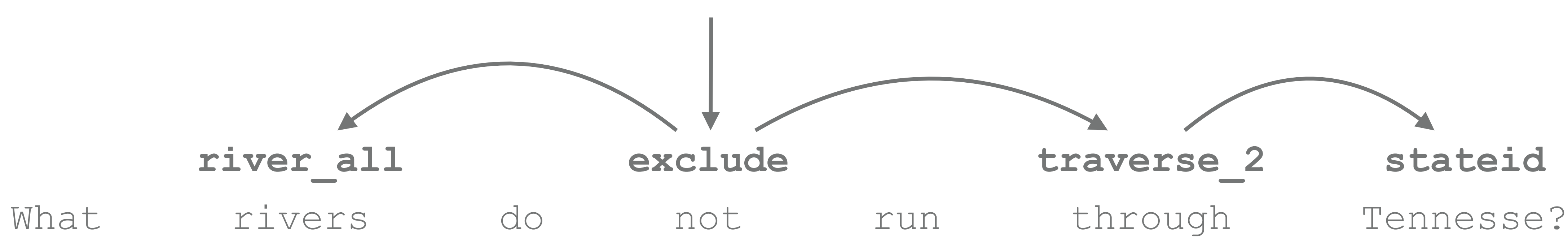
```
exclude ( river_all , traverse_2 ( stateid('Tennesse') ) )
```



Graph-based prediction

Joint tagging (entity+predicate) and parsing (argument identification)

- Non-spanning structure (function words, etc)
- Valency constraints
- Non-projective structure



GRAPH-BASED SEMANTIC PARSING

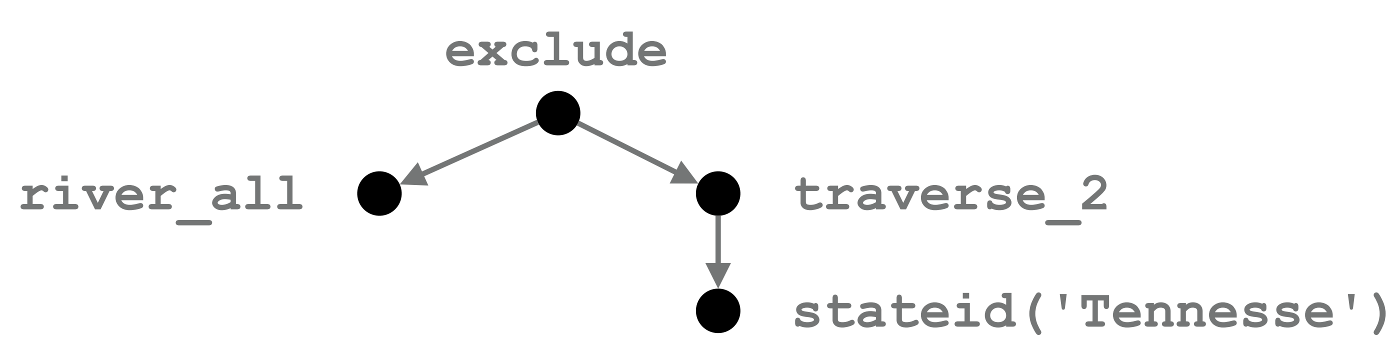
Semantic grammar

A semantic grammar is a tuple $\mathcal{G} = \langle\, E, T, f_{type}, f_{args} \,\rangle$ where:

- E is a set of predicates and entities (set of tags)
- T is a set of type
- $f_{type} : E \to T$ is a typing function that assigns a type to each tag
- $f_{args} : E \times T \to \mathbb{N}$ is a valency function that assigns the numbers of expected arguments of a given type

AST recognition

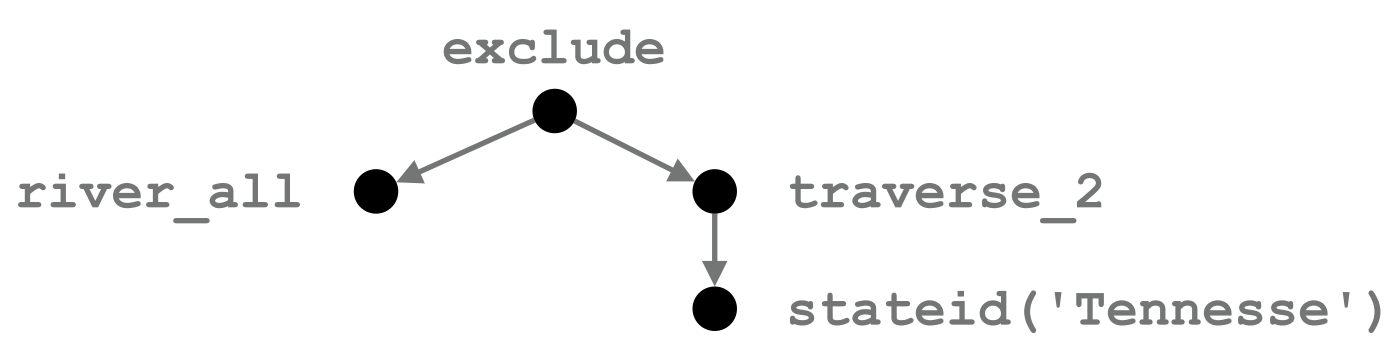
A labeled graph is a valid AST if and only if it can be recognized by the grammar $\mathcal{G}$



GRAPH-BASED SEMANTIC PARSING

Example

$E = \{exclude, river\_all, traverse\_1, traverse\_2, state\_id, \dots\}$ $T = \{river, state, \dots\}$



GRAPH-BASED SEMANTIC PARSING

Example

$E = \{exclude, river\_all, traverse\_1, traverse\_2, state\_id, \dots\}$ $T = \{river, state, \dots\}$

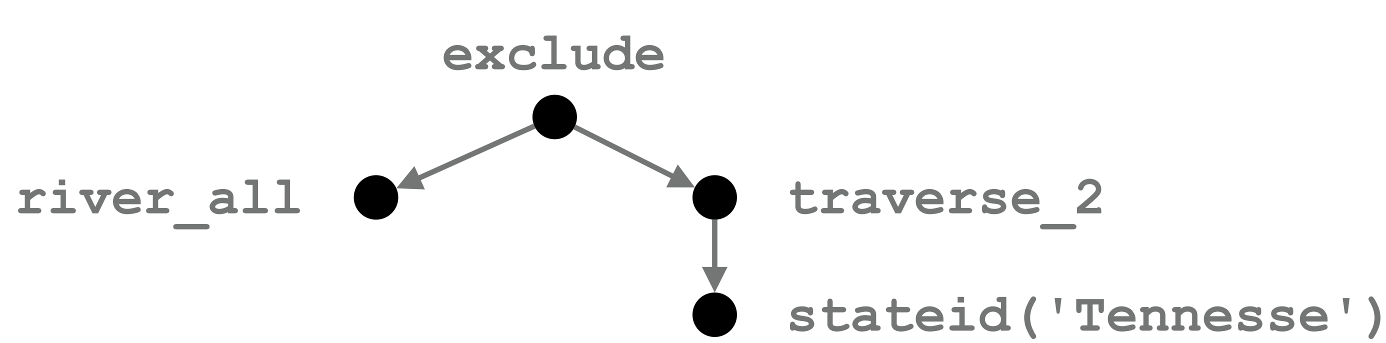
$$f_{type}(\ river\_all) = river$$

$$f_{type}(\ state\_id\ ) = state$$

$$f_{type}(\ traverse\_2\ ) = river$$

$$f_{type}(\ exclude\ ) = river$$

$$\dots$$



GRAPH-BASED SEMANTIC PARSING

Example

$E = \{exclude, river\_all, traverse\_1, traverse\_2, state\_id, \dots\}$

$T = \{river, state, \dots\}$

$f_{type}(\ river\_all) = river$

$f_{type}(\ state\_id\ ) = state$

$f_{type}(\ traverse\_2\ ) = river$

$f_{type}(\ exclude\ ) = river$

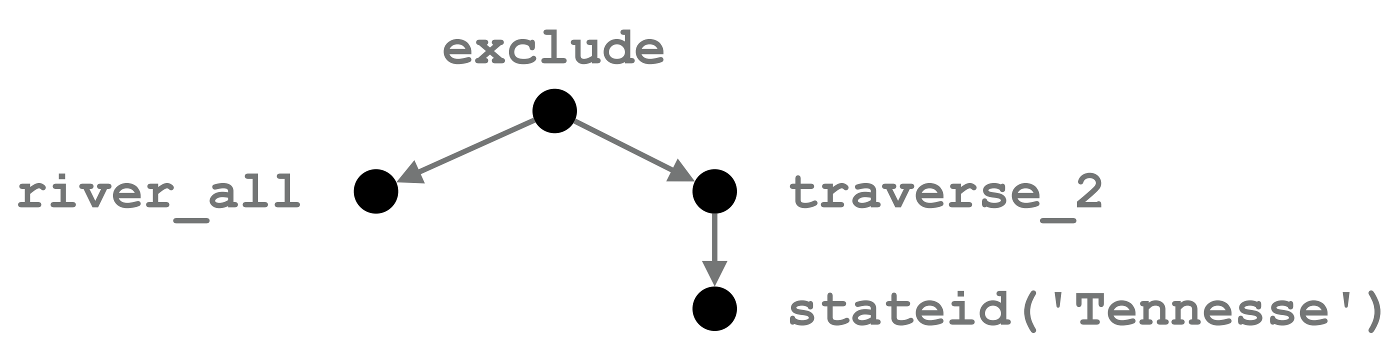
$\dots$

$f_{args}(\ river\_all, \dots) = 0$

$f_{args}(\ state\_id, \dots) = 0$

Entities

For all types



GRAPH-BASED SEMANTIC PARSING

Example

$E = \{exclude, river\_all, traverse\_1, traverse\_2, state\_id, \ldots\}$

$T = \{river, state, \ldots\}$

$f_{type}(\ river\_all) = river$

$f_{type}(\ state\_id\ ) = state$

$f_{type}(\ traverse\_2\ ) = river$

$f_{type}(\ exclude\ ) = river$

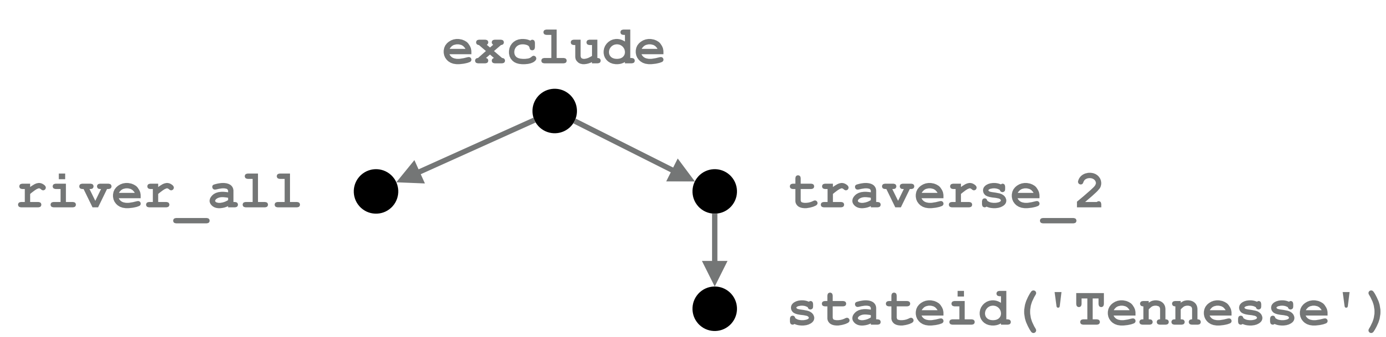
$\ldots$

$f_{args}(\ river\_all, \ldots\ ) = 0$

$f_{args}(\ state\_id, \ldots\ ) = 0$

$f_{args}(\ traverse\_2,\ river\ ) = 0$

$f_{args}(\ traverse\_2,\ state\ ) = 1$



GRAPH-BASED SEMANTIC PARSING

Example

$E = \{exclude, river\_all, traverse\_1, traverse\_2, state\_id, \ldots\}$ $\quad$ $T = \{river, state, \ldots\}$

$f_{type}(\ river\_all) = river$

$f_{type}(\ state\_id\ ) = state$

$f_{type}(\ traverse\_2\ ) = river$

$f_{type}(\ exclude\ ) = river$

$\ldots$

$f_{args}(\ river\_all, \ldots) = 0$

$f_{args}(\ state\_id, \ldots) = 0$

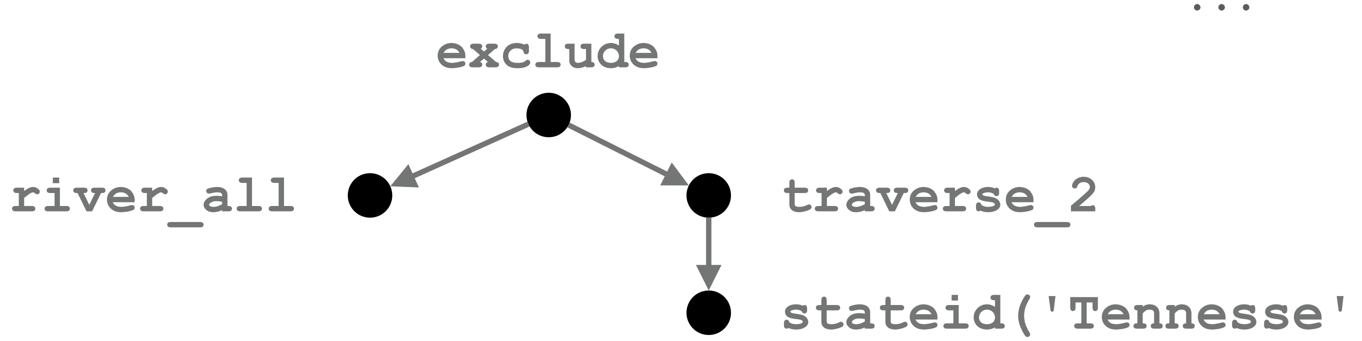
$f_{args}(\ traverse\_2,\ river\ ) = 0$

$f_{args}(\ traverse\_2,\ state\ ) = 1$

$f_{args}(\ exclude,\ river\ ) = 2$

$f_{args}(\ exclude,\ state\ ) = 0$

$\ldots$



GRAPH-BASED SEMANTIC PARSING

Example

$E = \{exclude, river\_all, traverse\_1, traverse\_2, state\_id, \dots\}$

$T = \{river, state, \dots\}$

$f_{type}(\ river\_all) = river$

$f_{type}(\ state\_id\ ) = state$

$f_{type}(\ traverse\_2\ ) = state$

$f_{type}(\ exclude\ ) = river$

$\dots$

$f_{args}(\ river\_all, \dots) = 0$

$f_{args}(\ state\_id, \dots) = 0$

$f_{args}(\ traverse\_2,\ river\ ) = 0$

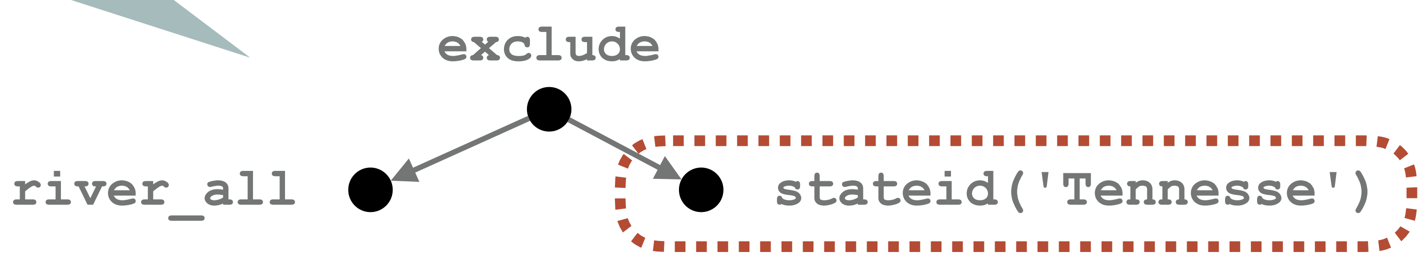
$f_{args}(\ traverse\_2,\ state\ ) = 1$

$f_{args}(\ exclude,\ river\ ) = 2$

$f_{args}(\ exclude,\ state\ ) = 0$

$\dots$

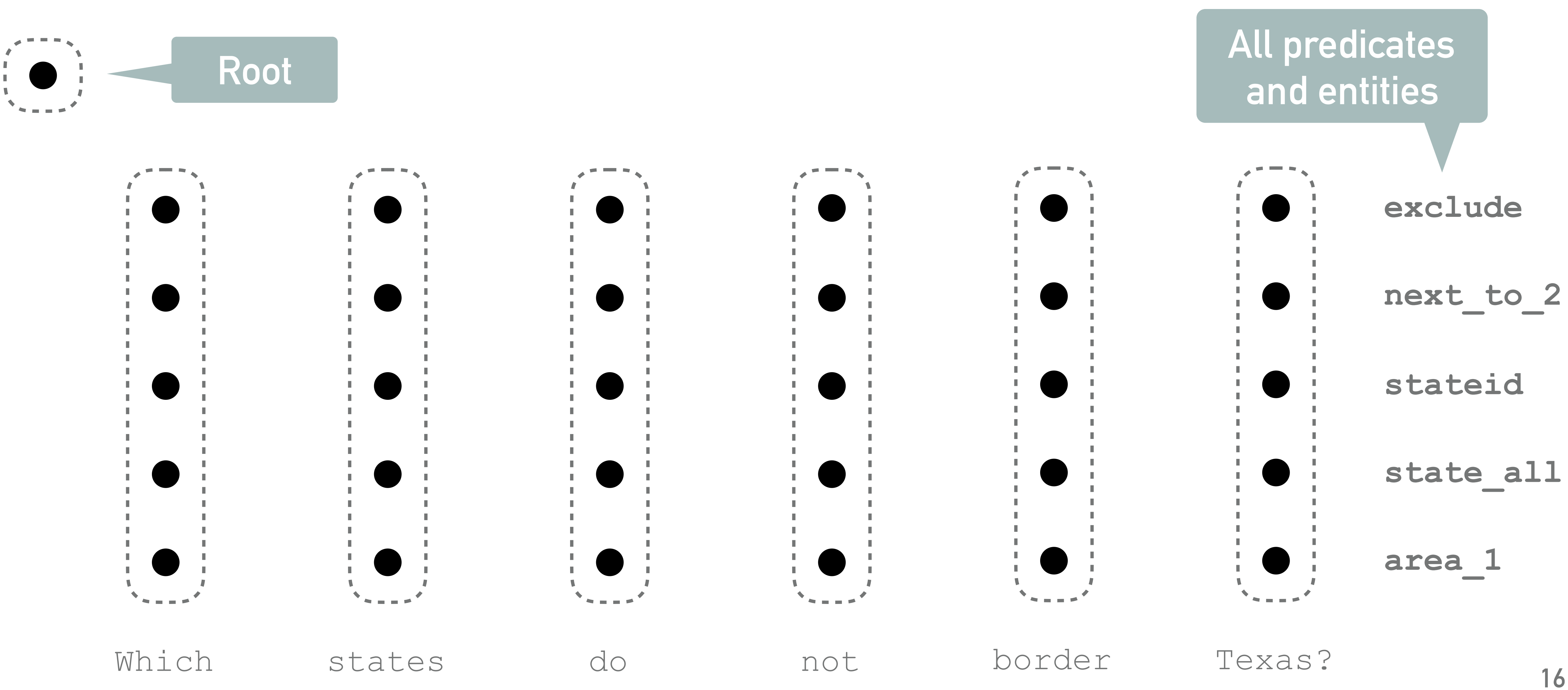
Invalid AST
for this grammar!



REDUCTION TO A GRAPH PROBLEM

Graph construction

1. For each word, create a cluster
2. In each cluster, create one vertex per element of T
3. Add all possible arcs (with weights from the neural network)

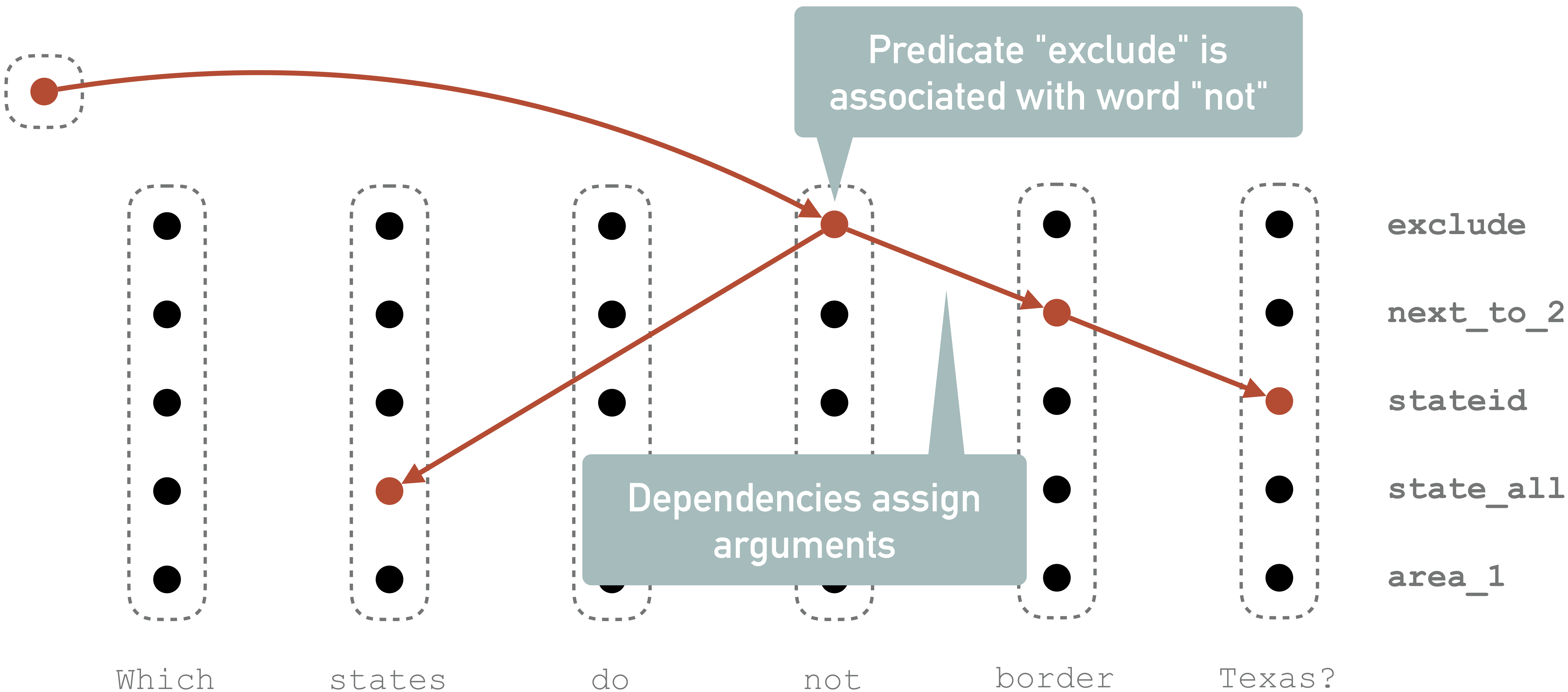


REDUCTION TO A GRAPH PROBLEM

AST parsing

Compute the rooted arborescence of maximum weight such that:

- There is at most one incident vertex per cluster
- Valency constraints are satisfied



NP-HARDNESS

AST parsing

Compute the rooted arborescence of maximum weight such that:

- There is at most one incident vertex per cluster
- Valency constraints are satisfied

Issue

This problem is NP hard! :(
(proof: by reduction of the maximum not-necessarily spanning arborescence problem)

NP-HARDNESS

AST parsing

Compute the rooted arborescence of maximum weight such that:

- There is at most one incident vertex per cluster
- Valency constraints are satisfied

Issue

This problem is NP hard! :(
(proof: by reduction of the maximum not-necessarily spanning arborescence problem)

Approximate solver

1. Formulation as a integer linear program
2. Relaxation of the integrality constraint
3. Identifying the difficult constraints
   and add them as penalties in the objective
4. Custom optimization algorithm based on the problem structure
   (indicator function smoothing + Frank-Wolfe)

$$\max_{\mathbf{z}\in[0,1]^d} \quad \langle \mathbf{z}, \phi \rangle$$
$$s.t. \quad z \in \mathscr{C}^{(easy)}$$
$$z \in \mathscr{C}^{(hard)}$$

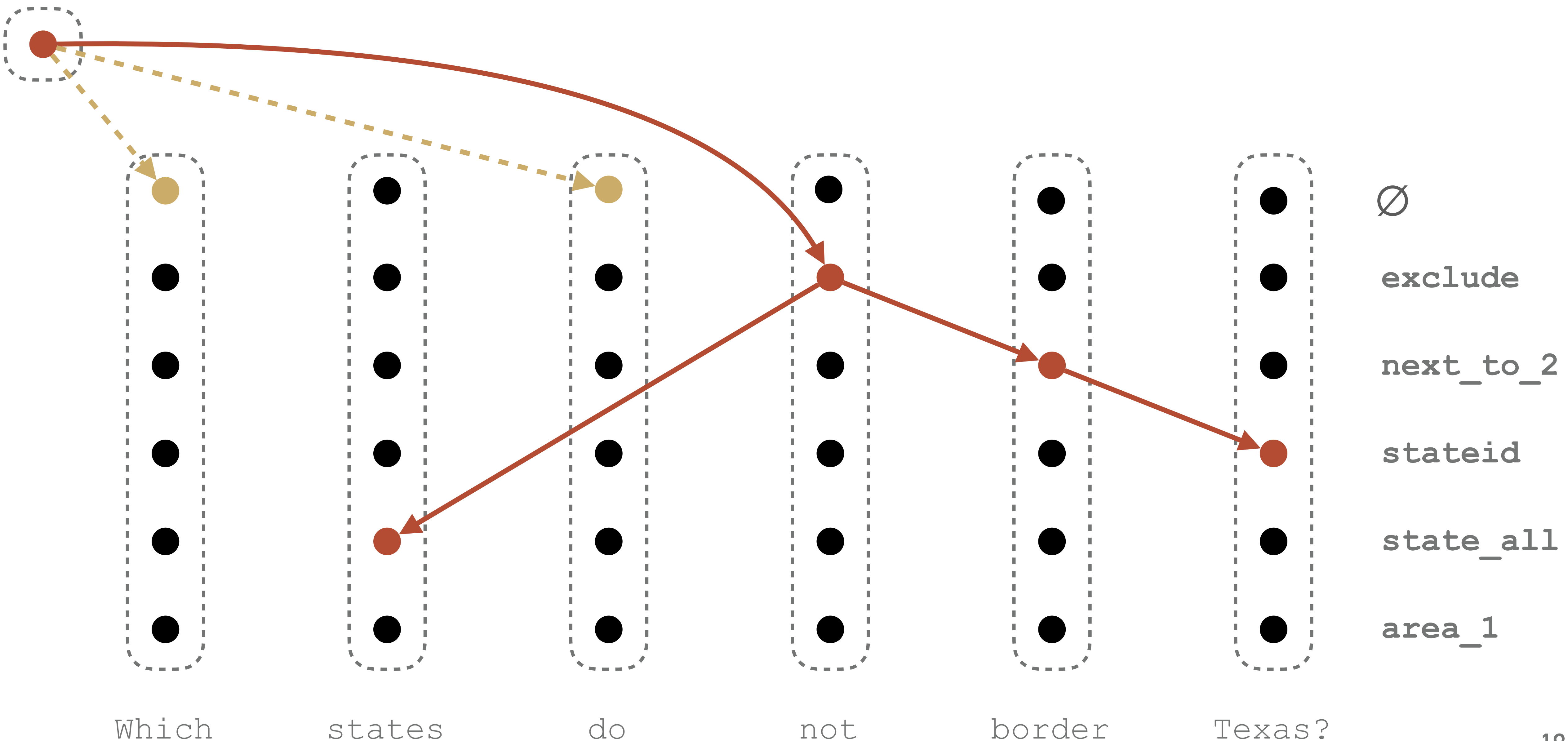
Valency constraints!

ALGORITHME INTUITION

Problem reformulation

To simplify the algorithm, we add "empty entities":

- The root must have exactly one outgoing arc to a non-empty entity/predicate
- The "empty entities" cannot have outgoing arcs in a solution



ALGORITHME INTUITION

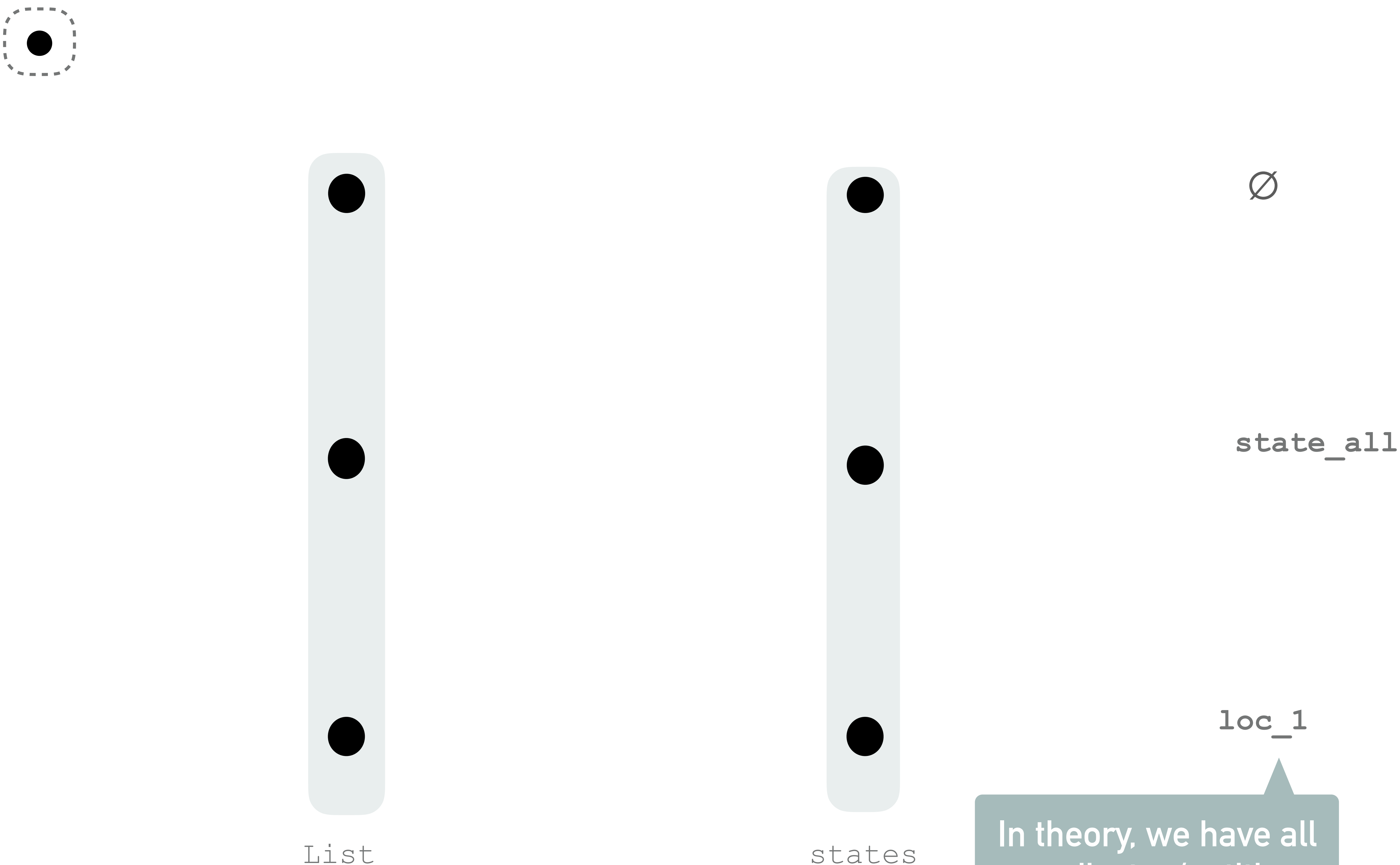
Input sentence

List

states

ALGORITHME INTUITION

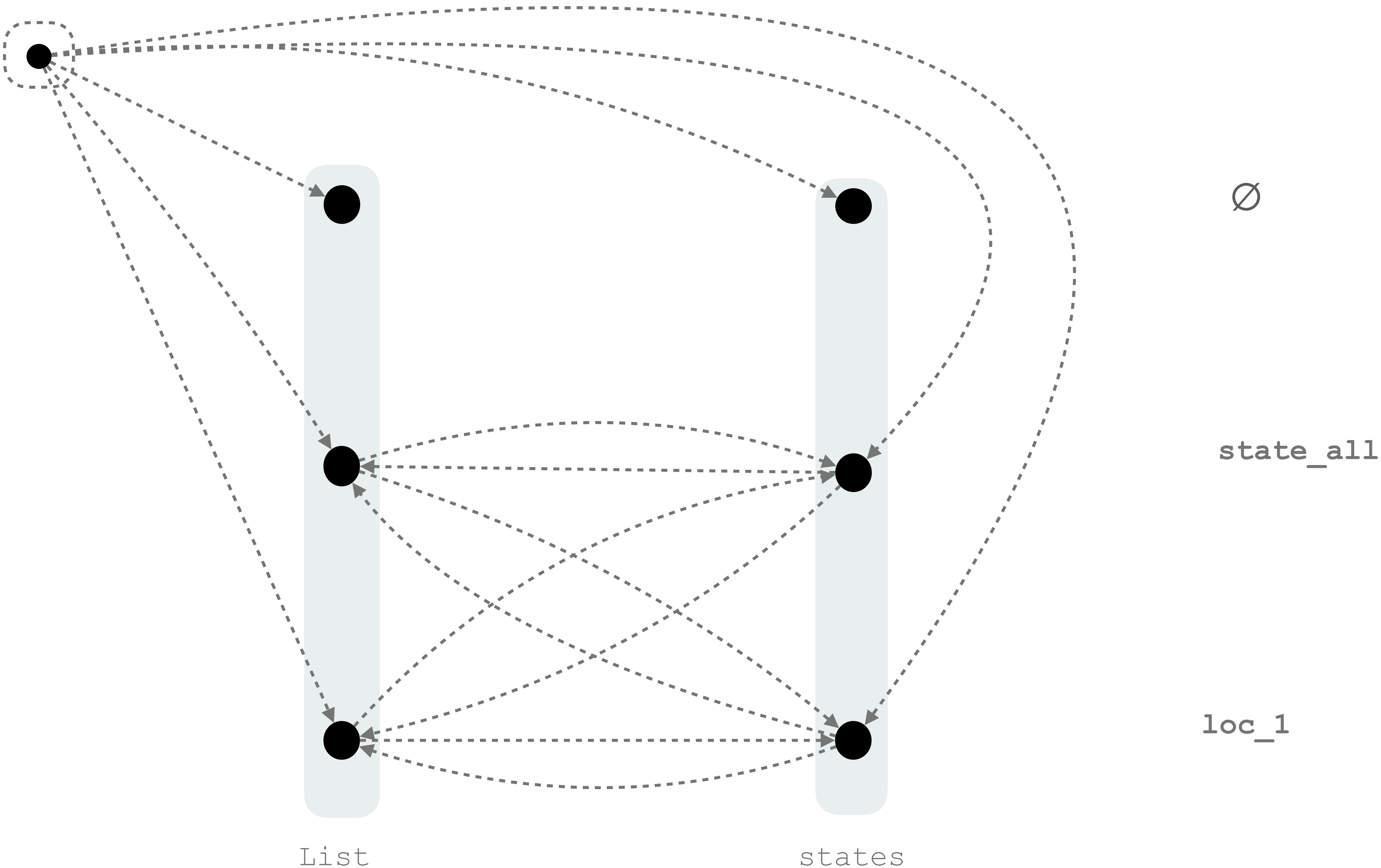
Create vertices



In theory, we have all predicates/entities

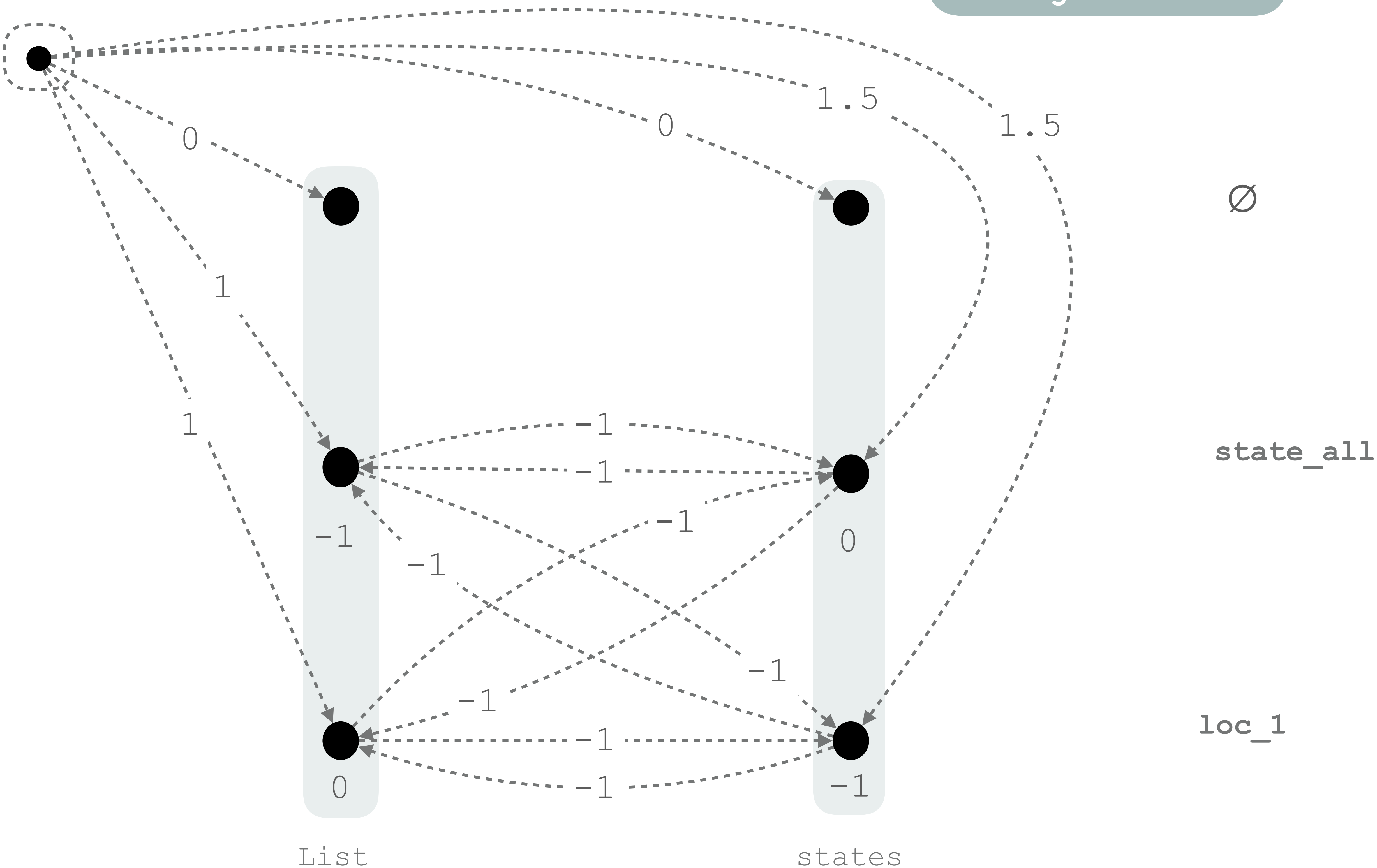
ALGORITHME INTUITION

Create arcs

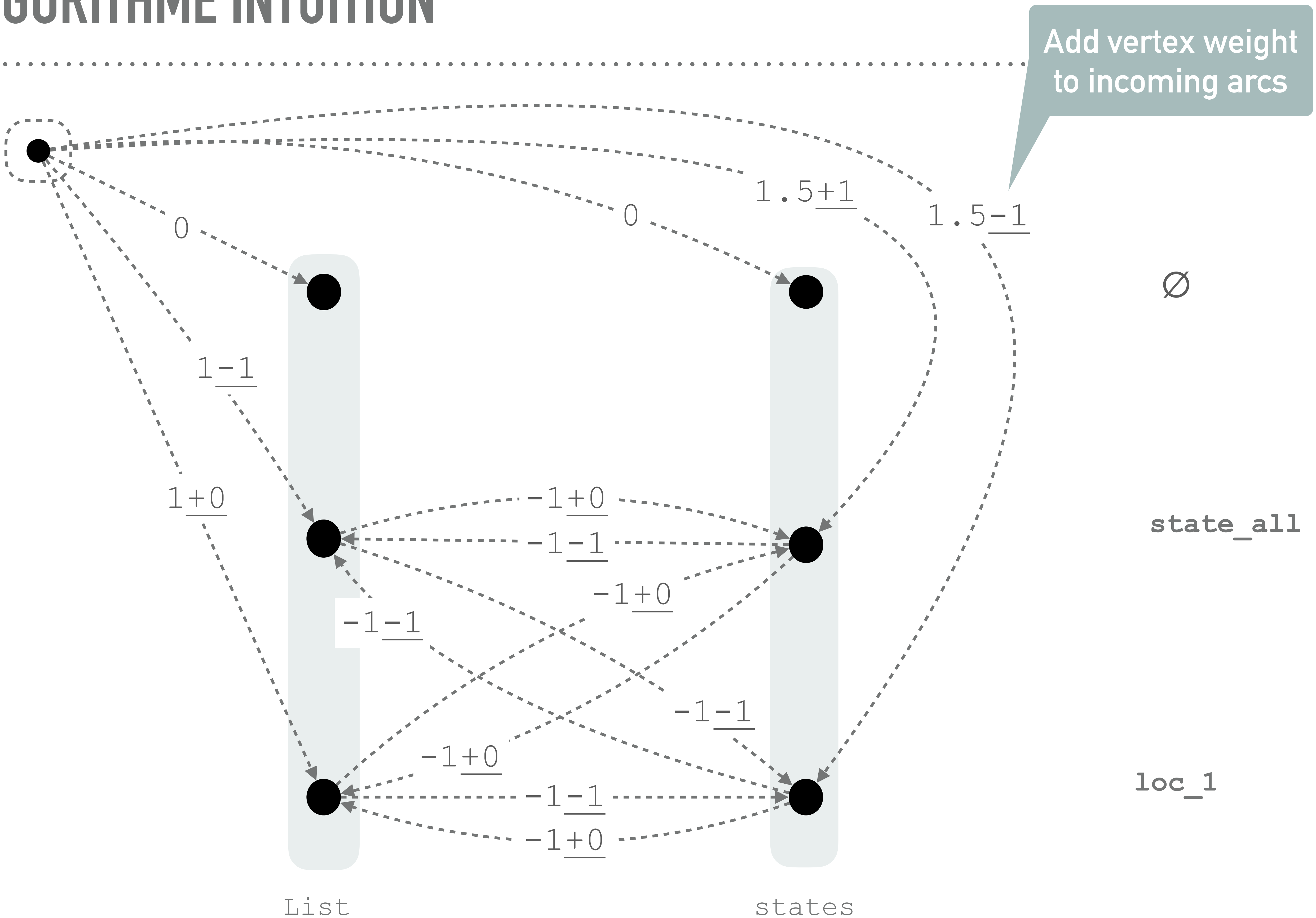


ALGORITHME INTUITION

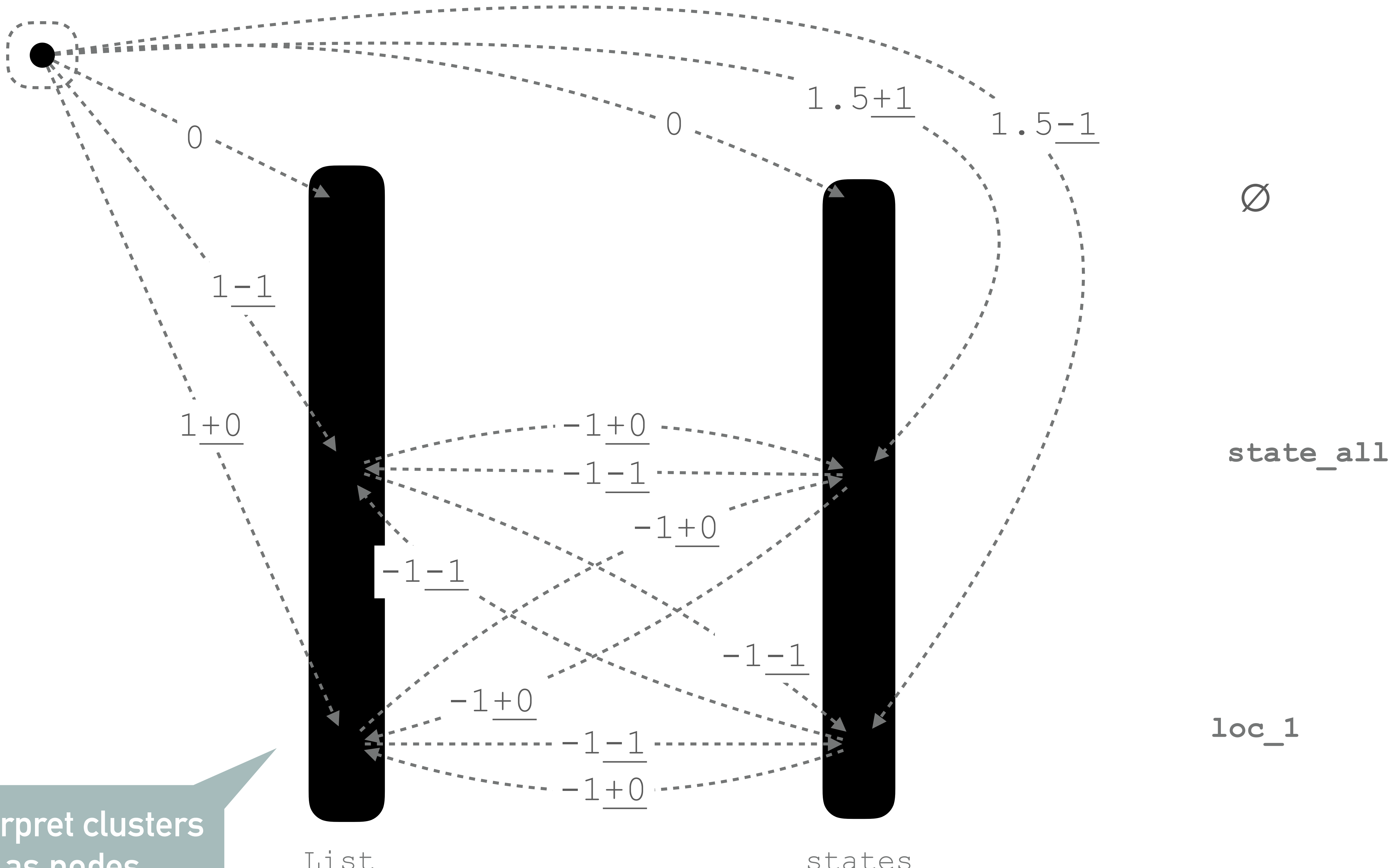
Create weights using the neural



ALGORITHME INTUITION



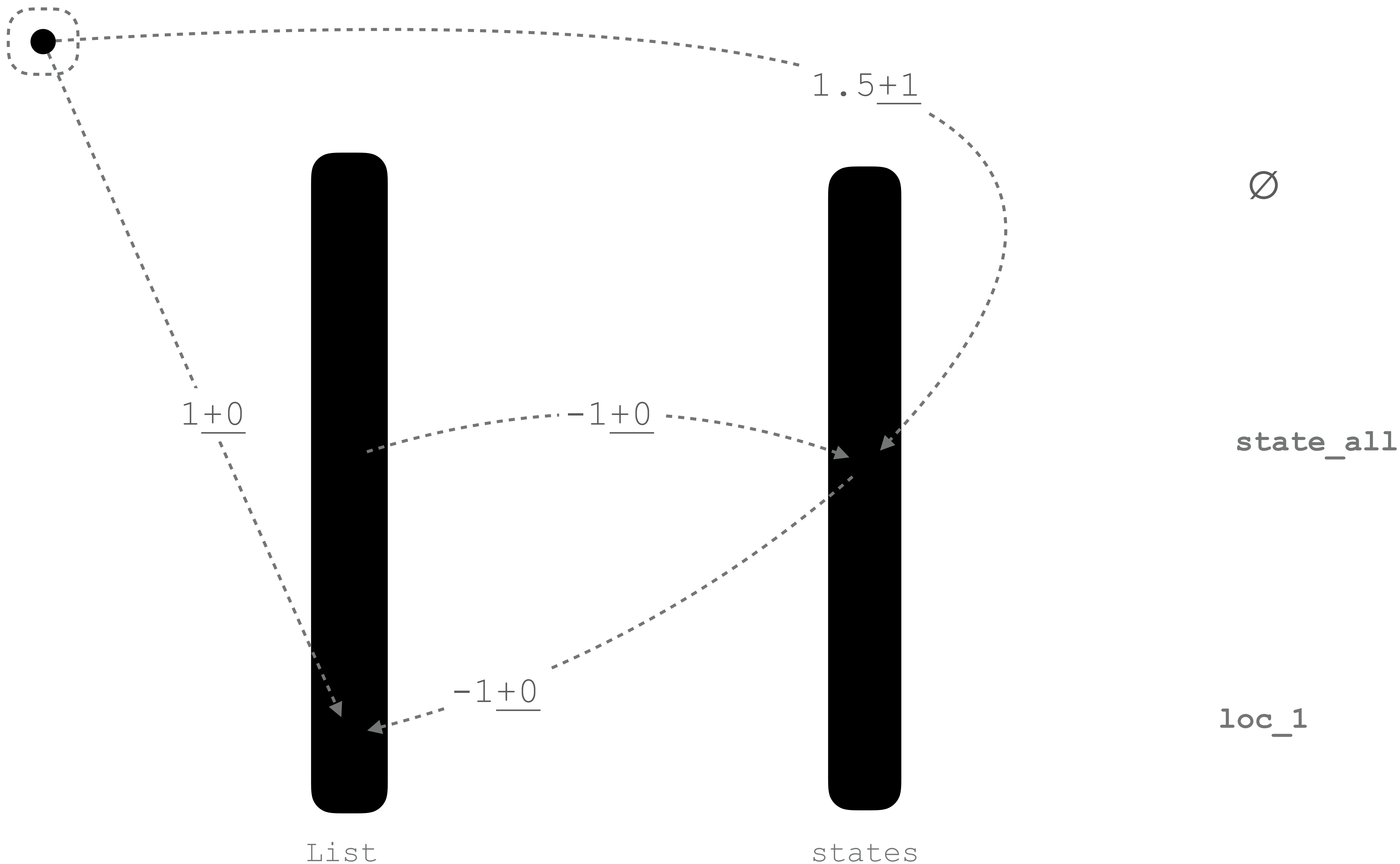
ALGORITHME INTUITION



Interpret clusters as nodes

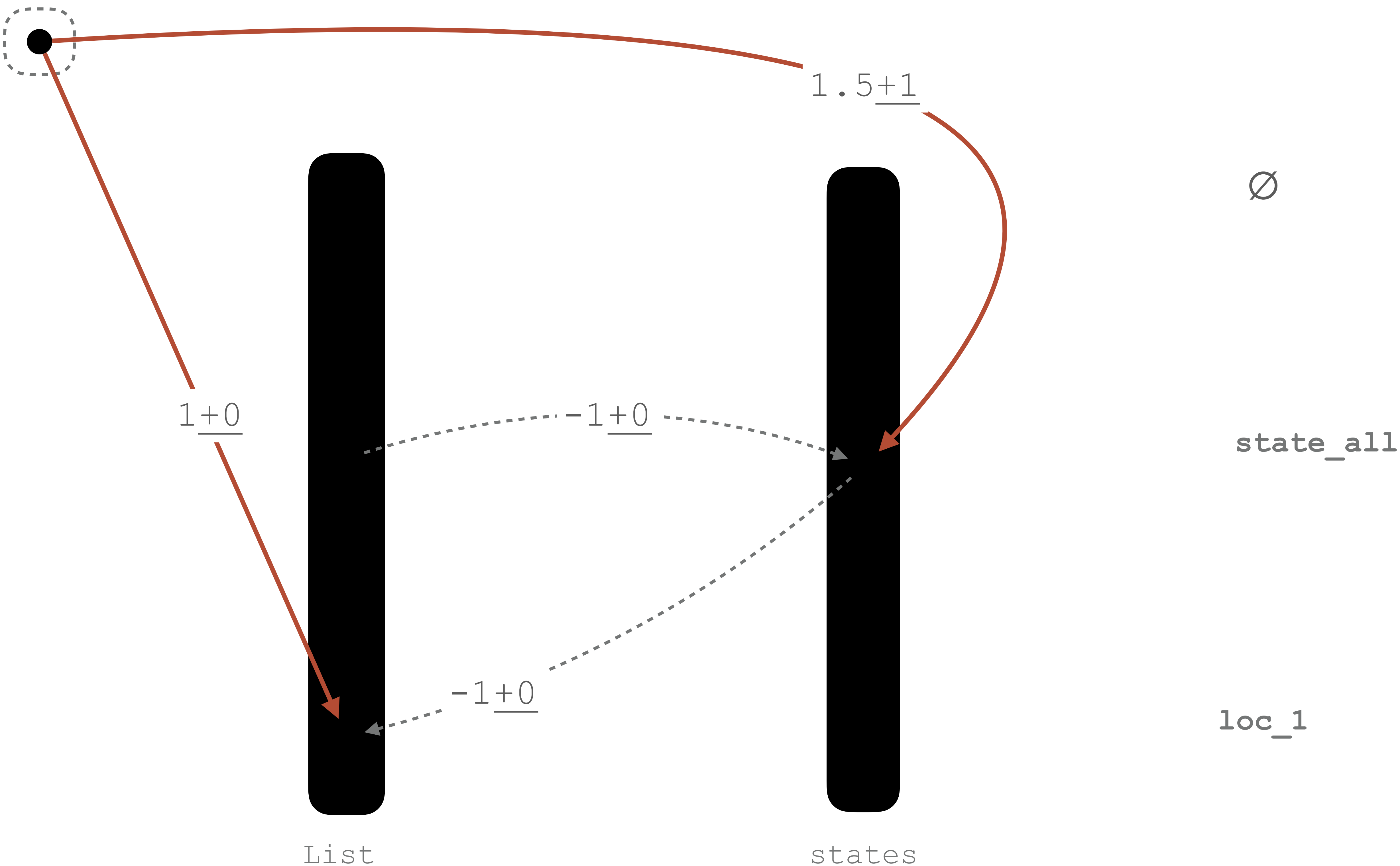
ALGORITHME INTUITION

Remove parallel arcs



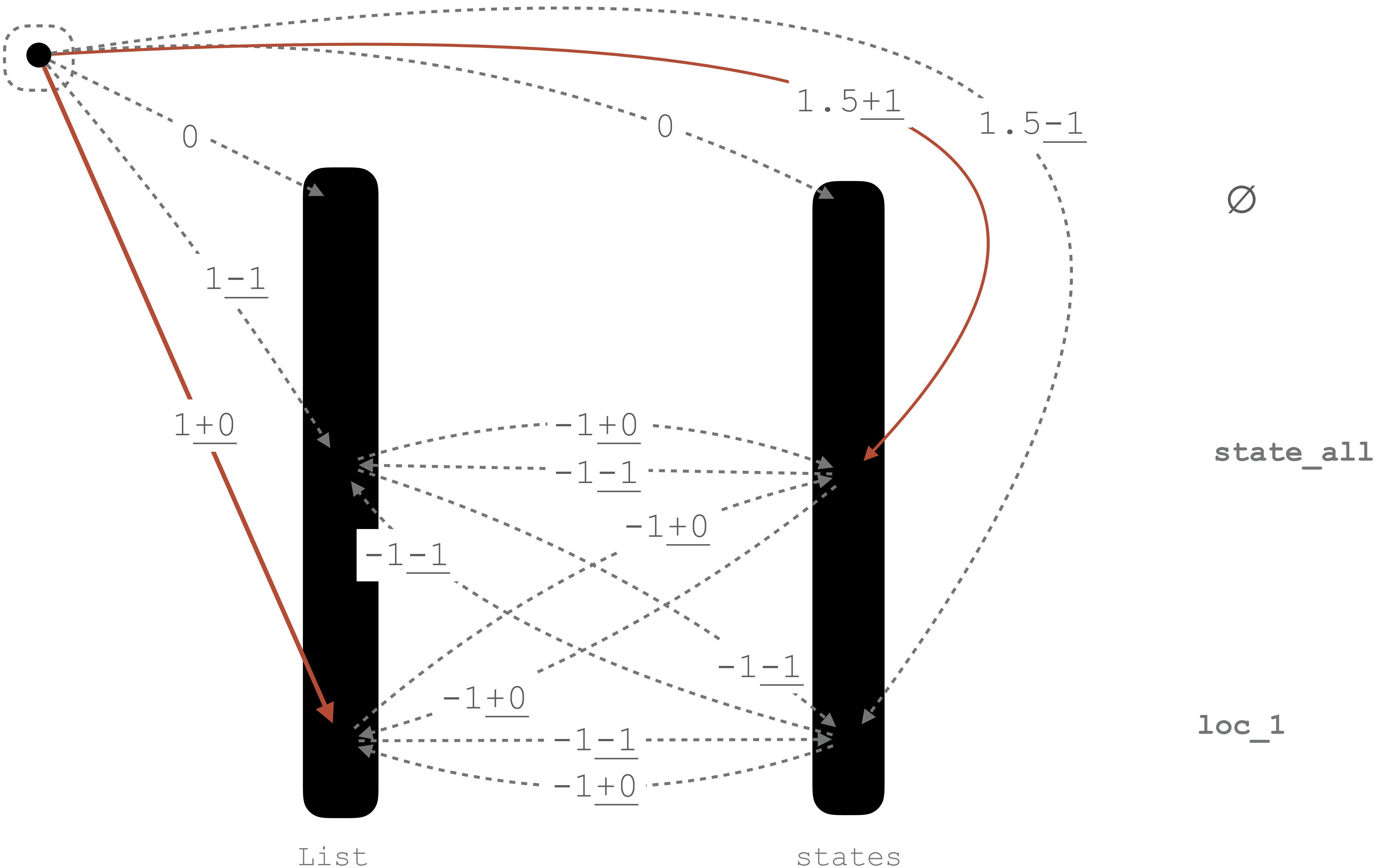
ALGORITHME INTUITION

Compute the maximum spanning arborescence over clusters

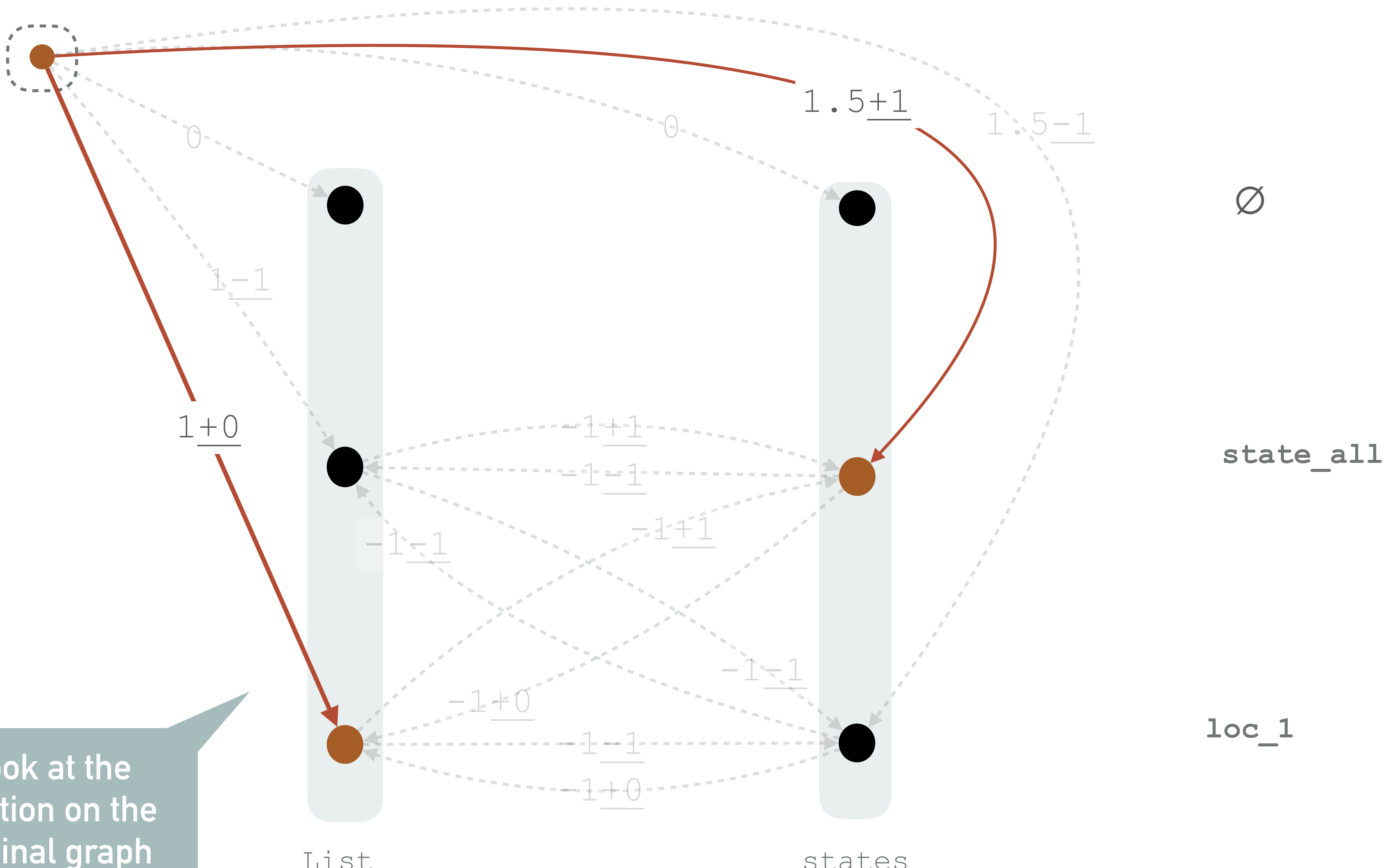


ALGORITHME INTUITION

Reconstruct full graph

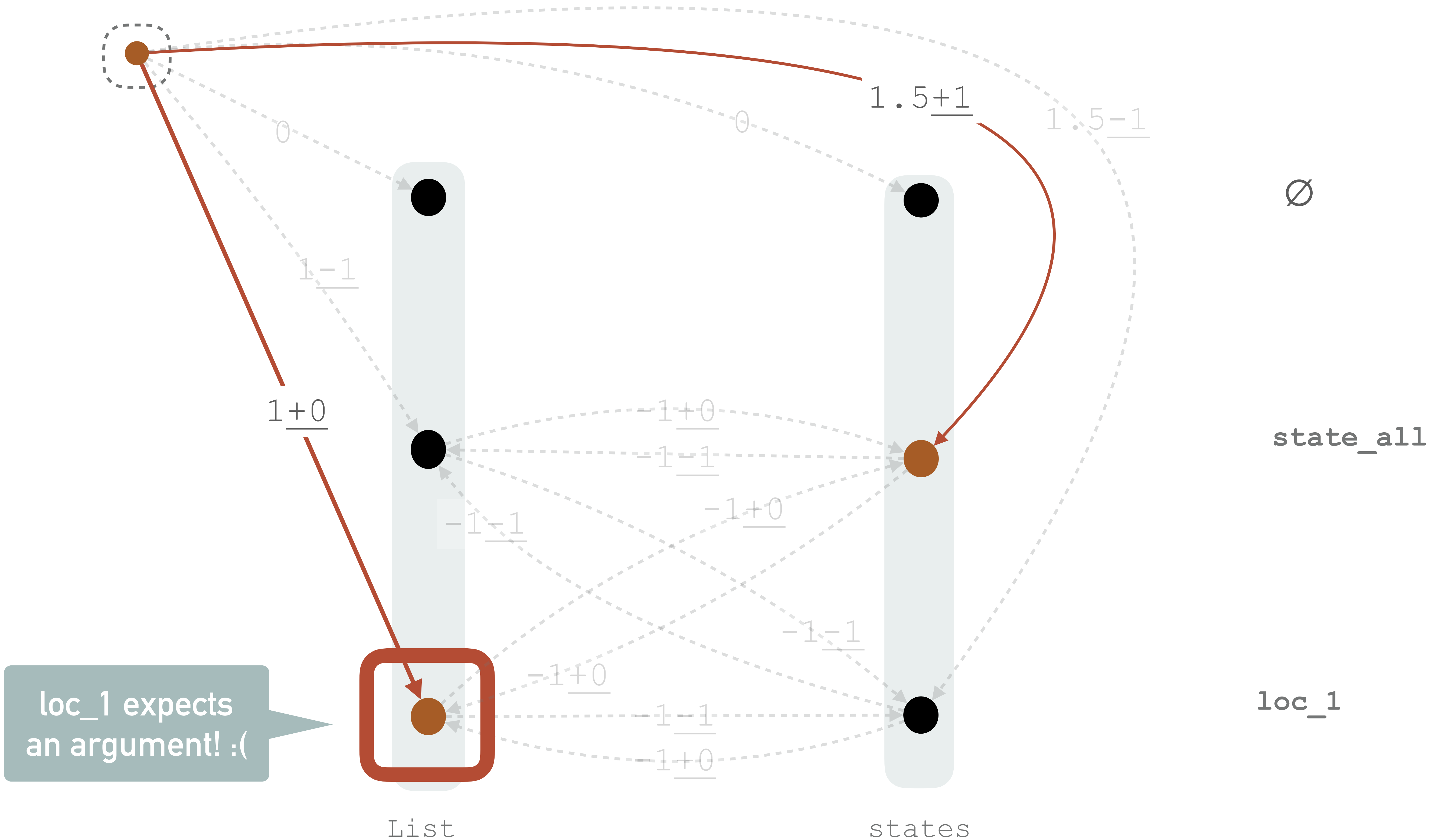


ALGORITHME INTUITION

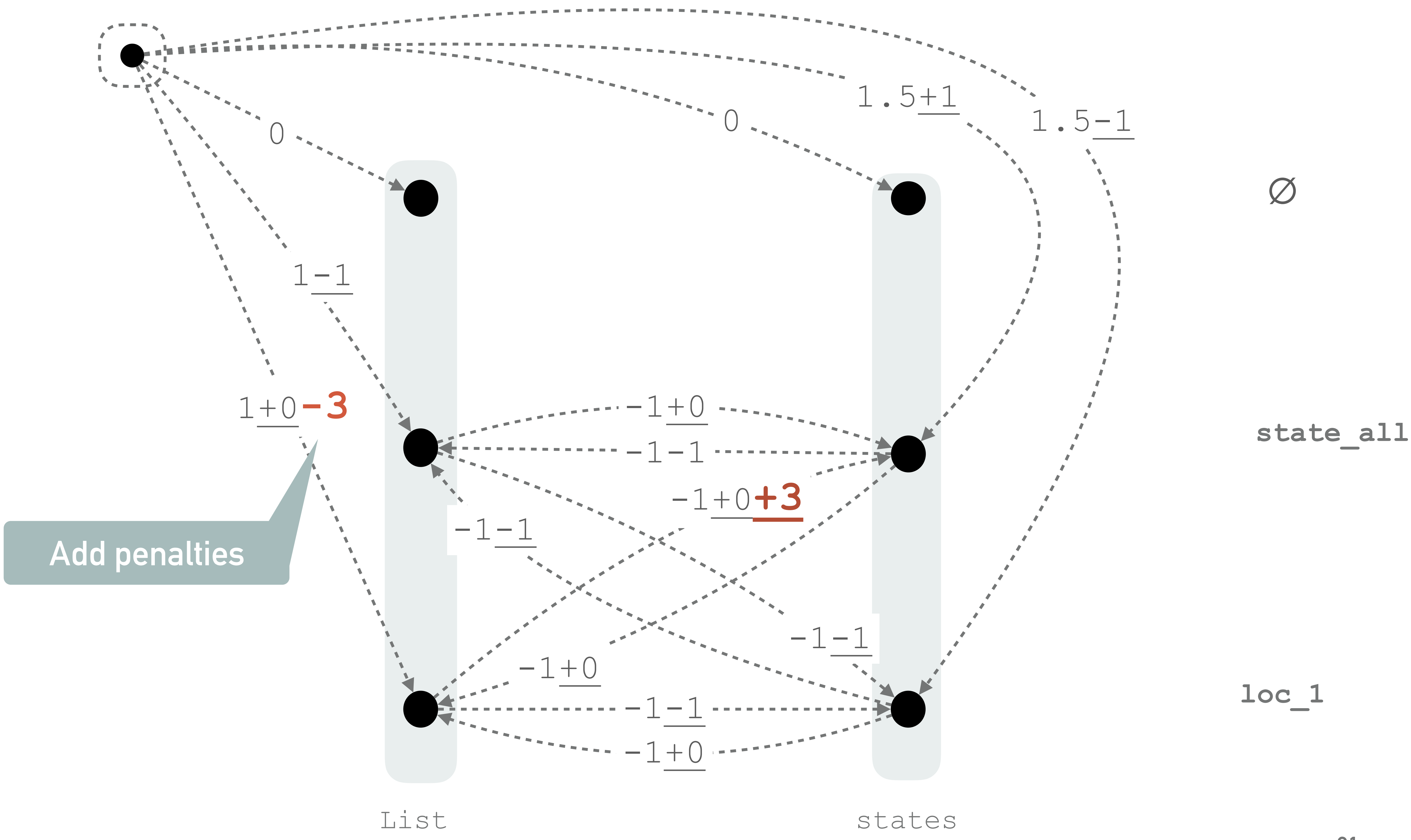


Look at the solution on the original graph

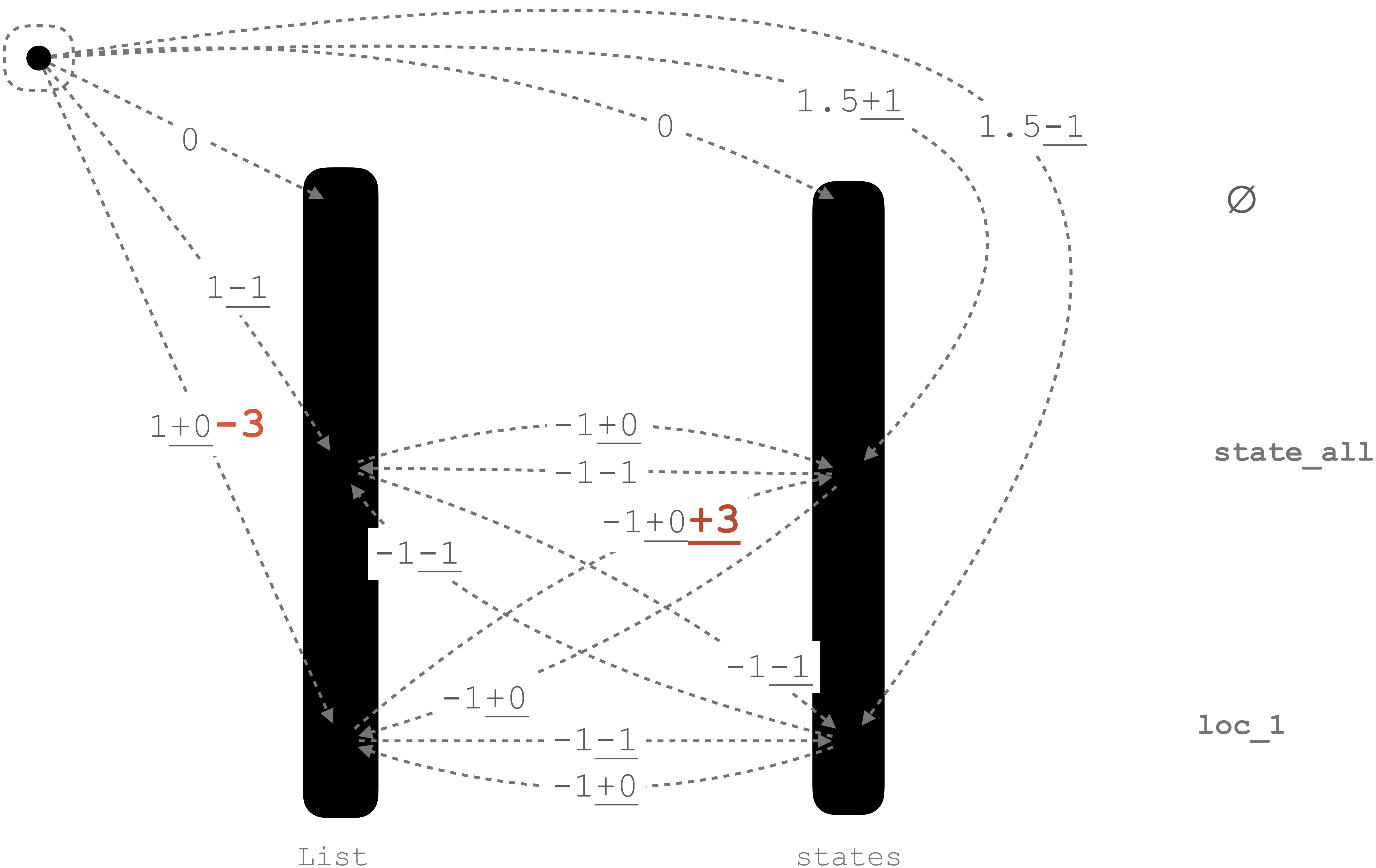
ALGORITHME INTUITION



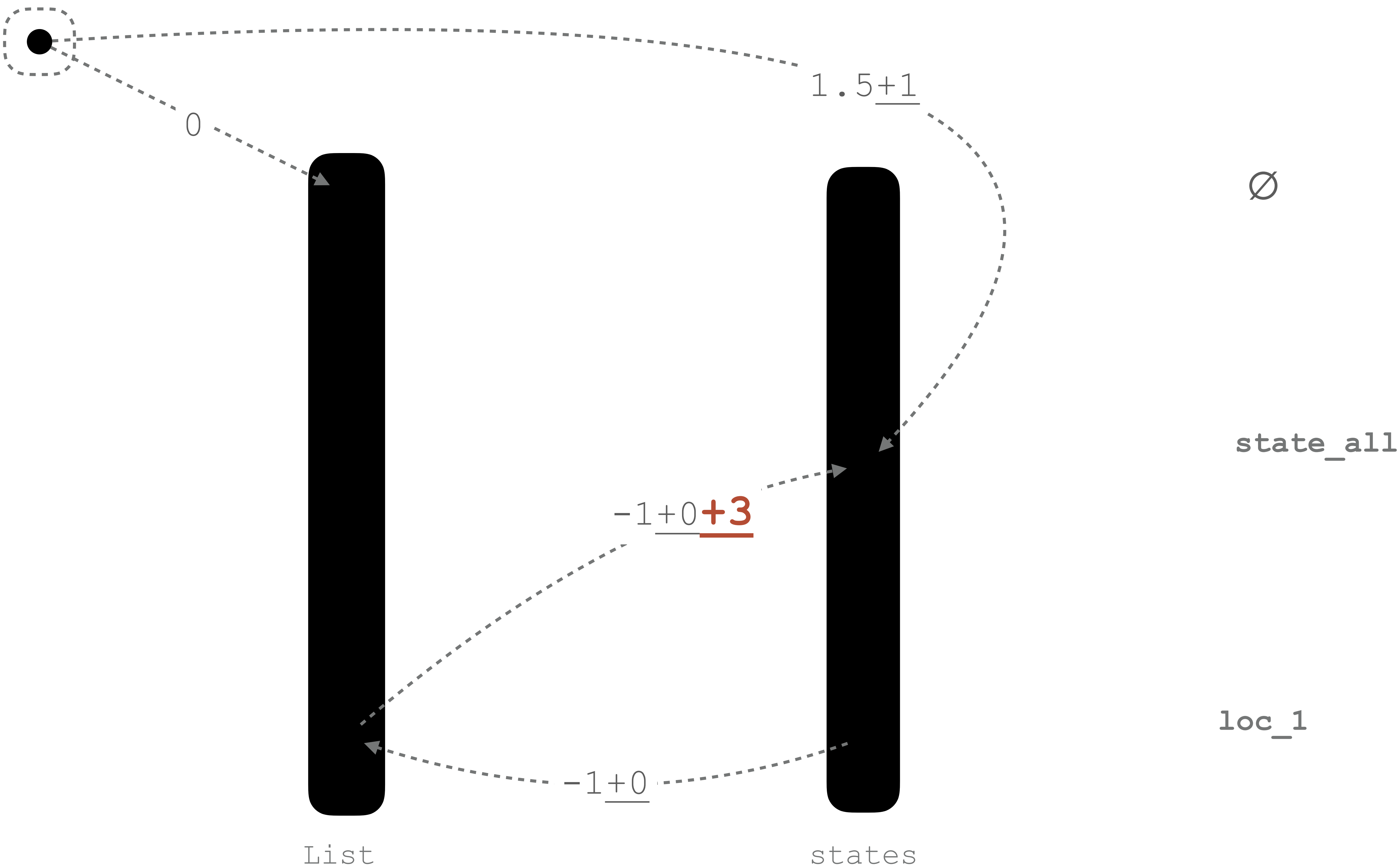
ALGORITHME INTUITION



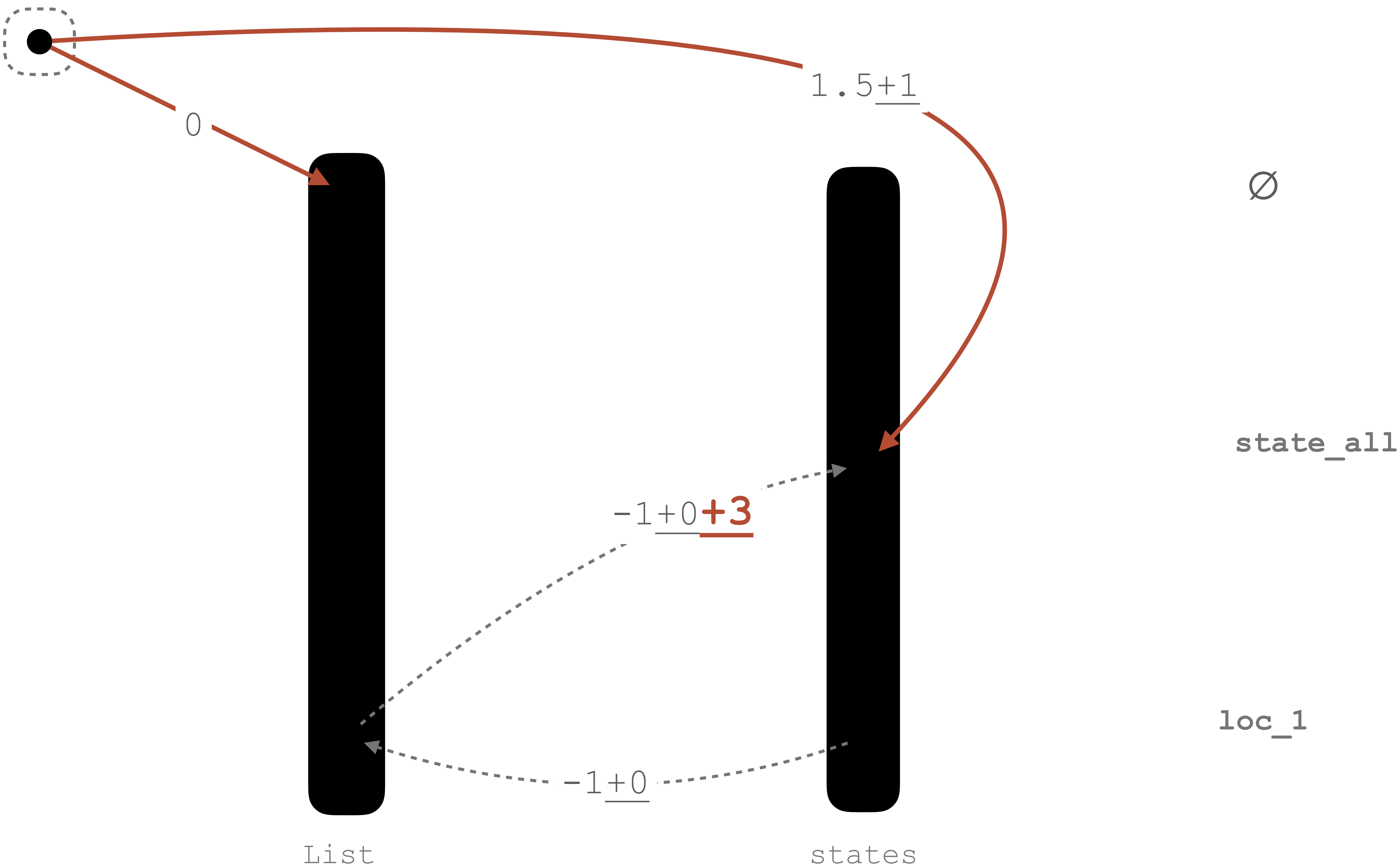
ALGORITHME INTUITION



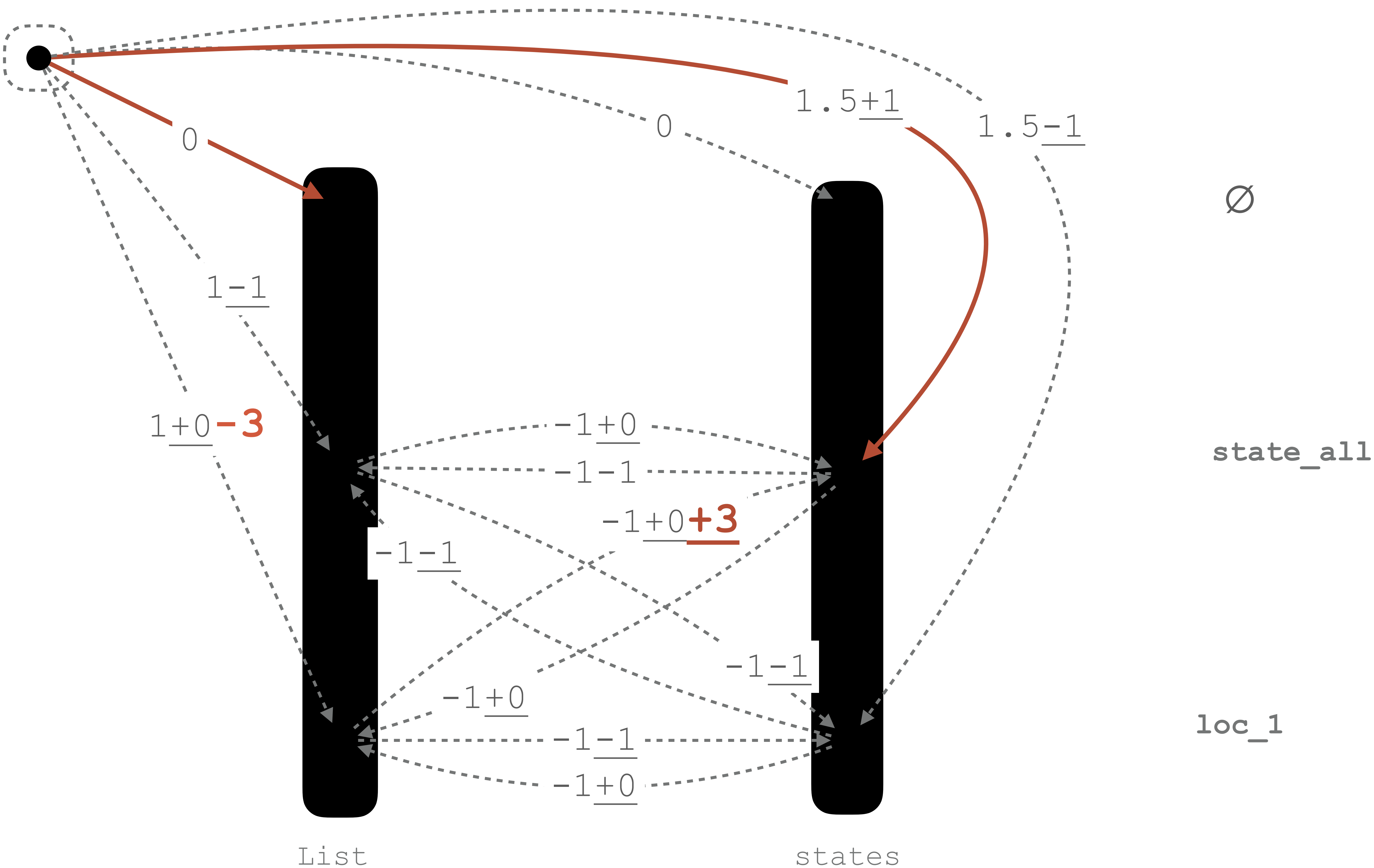
ALGORITHME INTUITION



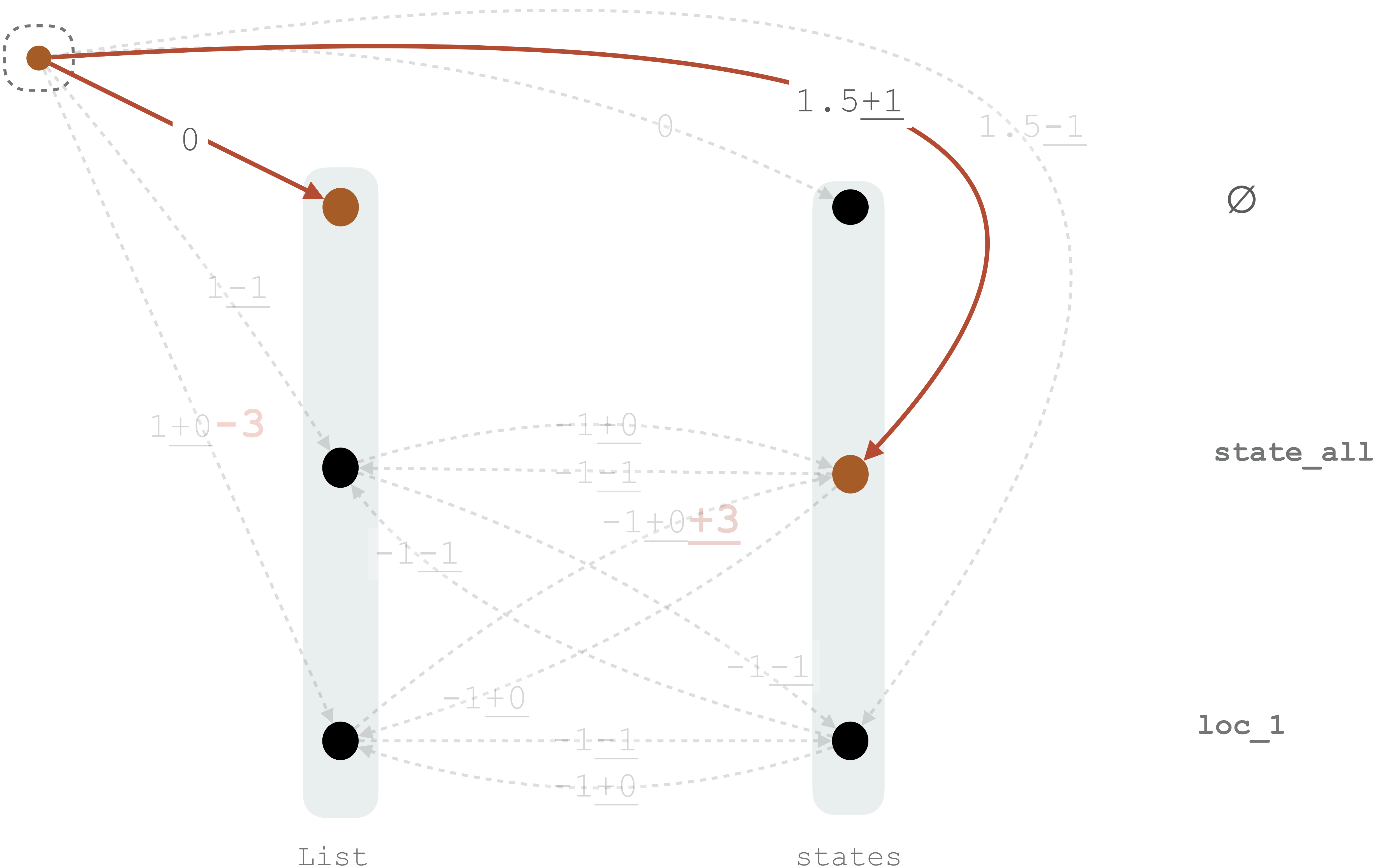
ALGORITHME INTUITION



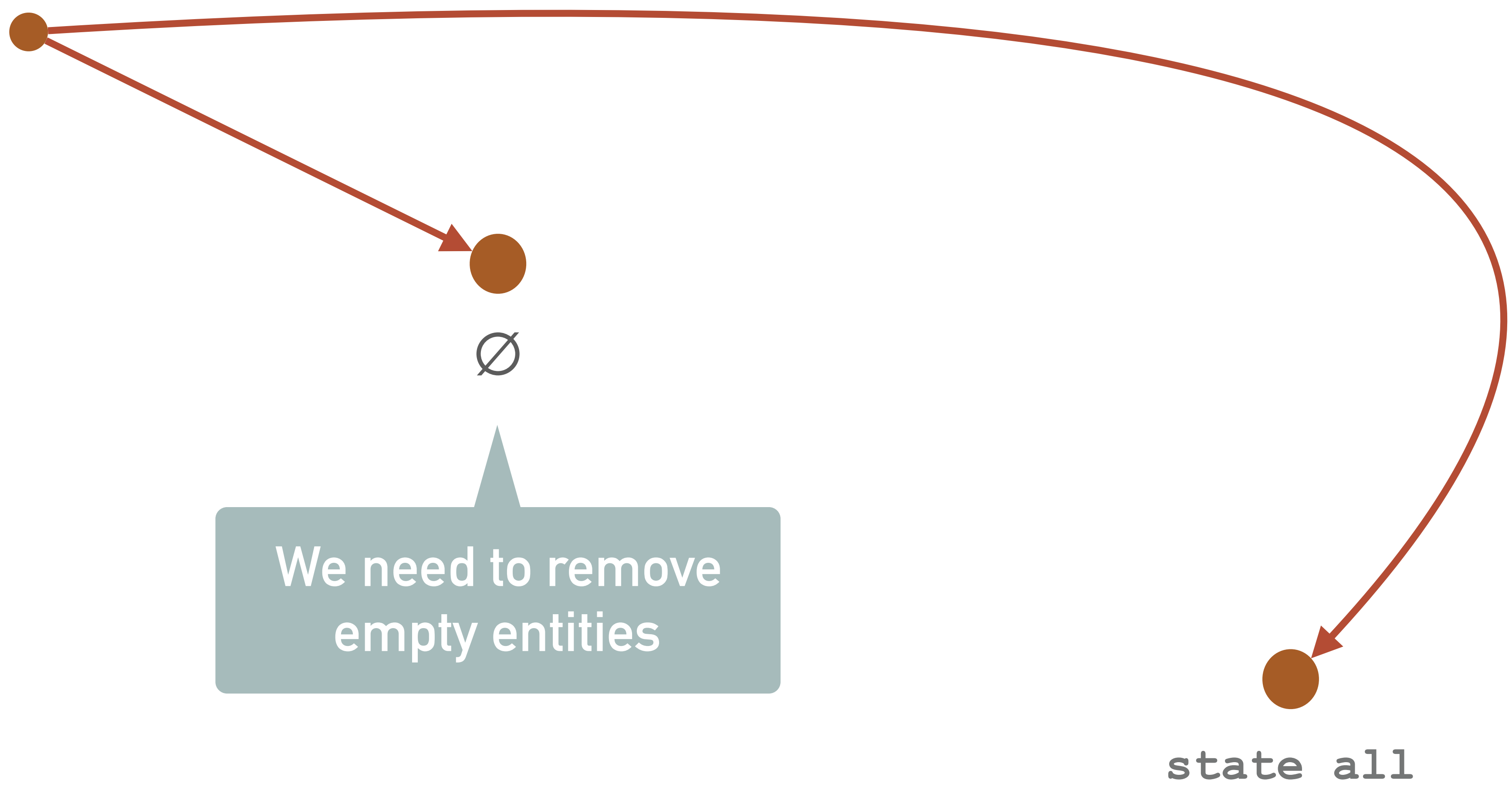
ALGORITHME INTUITION



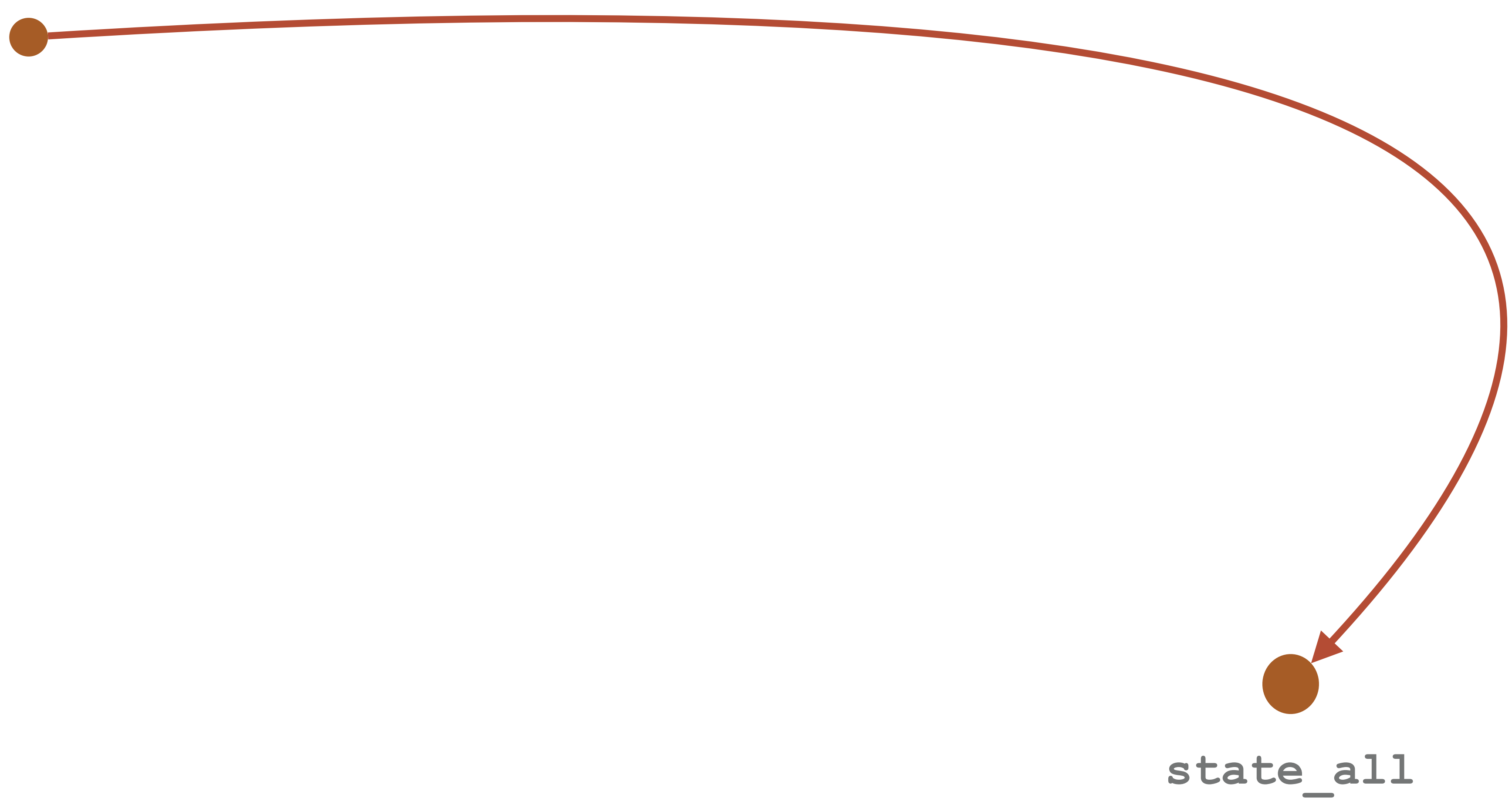
ALGORITHME INTUITION



ALGORITHME INTUITION



ALGORITHME INTUITION



This is a valid AST!

SUPERVISED LEARNING

NEGATIVE LOG-LIKELIHOOD

Notations

- Search space: directed graph $G = (V, A)$ where V is the set of vertices and $A \subseteq V \times V$ is the set of arcs
- Vertex selection vector: $\mathbf{x} \in \{0,1\}^V$
- Arc selection vector: $\mathbf{y} \in \{0,1\}^A$
- Set of feasible solution (i.e. set of ASTs): $(\mathbf{x}, \mathbf{y}) \in \mathscr{C}$

Weight vectors

- Vertex weights: $\mu \in \mathbb{R}^V$
- Arc weights: $\phi \in \mathbb{R}^A$

NEGATIVE LOG-LIKELIHOOD

Notations

- Search space: directed graph $G = (V, A)$ where V is the set of vertices and $A \subseteq V \times V$ is the set of arcs
- Vertex selection vector: $\mathbf{x} \in \{0,1\}^V$
- Arc selection vector: $\mathbf{y} \in \{0,1\}^A$
- Set of feasible solution (i.e. set of ASTs): $(\mathbf{x}, \mathbf{y}) \in \mathscr{C}$

Weight vectors

- Vertex weights: $\mu \in \mathbb{R}^V$
- Arc weights: $\phi \in \mathbb{R}^A$

Boltzmann distribution over ASTs

Log-partition function

$$p_{\mu,\phi}(\mathbf{x}, \mathbf{y}) = \begin{cases} \exp\left(\langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle - c(\mu, \phi) \right) & \textit{if } (\mathbf{x}, \mathbf{y}) \in \mathscr{C} \\ 0 & \textit{otherwise,} \end{cases}$$

where $$c(\mu, \phi) = \log \sum_{(\mathbf{x}', \mathbf{y}') \in \mathscr{C}} \exp\left(\langle \mu, \mathbf{x}' \rangle + \langle \phi, \mathbf{y}' \rangle \right)$$

NEGATIVE LOG-LIKELIHOOD

Boltzmann distribution over ASTs

Log-partition function

$$p_{\mu,\phi}(\mathbf{x},\mathbf{y}) = \begin{cases} \exp\left(\langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle - c(\mu,\phi) \right) & \textit{if } (\mathbf{x},\mathbf{y}) \in \mathscr{C} \\ 0 & \textit{otherwise,} \end{cases}$$

where $$c(\mu,\phi) = \log \sum_{(\mathbf{x}',\mathbf{y}')\in\mathscr{C}} \exp\left(\langle \mu, \mathbf{x}' \rangle + \langle \phi, \mathbf{y}' \rangle \right)$$

Negative log-likelihood loss

(probably) intractable!

$$\begin{aligned} \ell(\mu,\phi;\mathbf{x},\mathbf{y}) &= -\log p_{\mu,\phi}(\mathbf{x},\mathbf{y}) \\ &= -\langle \mu, \mathbf{x} \rangle - \langle \phi, \mathbf{y} \rangle + c(\mu,\phi) \end{aligned}$$

! We cannot compute the loss function! :(

VARIATIONAL APPROXIMATION

Change of notation

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \qquad \theta = \begin{bmatrix} \mu \\ \phi \end{bmatrix} \qquad \mathscr{Z} = \{\ \mathbf{z}^{(1)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k)}\ \}$$

Set of feasible ASTs

VARIATIONAL APPROXIMATION

Change of notation

Set of feasible ASTs

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \qquad \theta = \begin{bmatrix} \mu \\ \phi \end{bmatrix} \qquad \mathscr{Z} = \{\ \mathbf{z}^{(1)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k)}\ \}$$

Upper bound on the log-partition function

$$c(\theta) = \log \sum_{\mathbf{z} \in \mathscr{Z}} \exp \langle \theta, \mathbf{z} \rangle$$

VARIATIONAL APPROXIMATION

Change of notation

$$\mathbf{z} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \qquad \theta = \begin{bmatrix}\mu\\ \phi\end{bmatrix} \qquad \mathscr{Z} = \{\ \mathbf{z}^{(1)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k)}\ \}$$

Set of feasible ASTs

Upper bound on the log-partition function

$$c(\theta) = \log \sum_{\mathbf{z}\in\mathscr{Z}} \exp\, \langle \theta, \mathbf{z}\rangle$$

$$= \underbrace{\max_{\mathbf{p}\in\triangle^k} \langle \mathbf{p}, \mathbf{U}\theta\rangle - \underbrace{\sum_i p_i \log p_i}_{=H[\mathbf{p}]}}_{\textit{Fenchel bi-conjugate}}$$

Each row is a feasible AST

$$\mathbf{U} = \begin{bmatrix} z_1^{(1)}, & z_1^{(1)}, & \ldots, & z_d^{(1)} \\ z_1^{(2)}, & z_1^{(2)}, & \ldots, & z_d^{(2)} \\ \vdots & & & \\ \vdots & & & \\ z_1^{(k)}, & z_1^{(k)}, & \ldots, & z_d^{(k)} \end{bmatrix}$$

$$\mathbf{U}\theta = \begin{bmatrix} \langle \mathbf{z}^{(1)}, \theta\rangle \\ \langle \mathbf{z}^{(2)}, \theta\rangle \\ \vdots \\ \vdots \\ \langle \mathbf{z}^{(k)}, \theta\rangle \end{bmatrix}$$

Weight of each AST

VARIATIONAL APPROXIMATION

Change of notation

Set of feasible ASTs

$$\mathbf{z} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \qquad \theta = \begin{bmatrix}\mu\\ \phi\end{bmatrix} \qquad \mathscr{Z} = \{\ \mathbf{z}^{(1)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k)}\ \}$$

Upper bound on the log-partition function

$$\begin{aligned} c(\theta) &= \log \sum_{\mathbf{z}\in\mathscr{Z}} \exp\,\langle \theta, \mathbf{z}\rangle \\ &= \max_{\mathbf{p}\in\triangle^k} \langle \mathbf{p}, \mathbf{U}\theta\rangle \ \underbrace{-\ \sum_i p_i \log p_i}_{=H[\mathbf{p}]} \\ &= \max_{\mathbf{p}\in\triangle^k} \underbrace{\left(\mathbf{p}^\top \mathbf{U}\right)}\theta \ +\ H[\mathbf{p}] \end{aligned}$$

Marginal distribution

VARIATIONAL APPROXIMATION

Change of notation

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \qquad \theta = \begin{bmatrix} \mu \\ \phi \end{bmatrix} \qquad \mathscr{Z} = \{\ \mathbf{z}^{(1)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k)}\ \}$$

Set of feasible ASTs

Upper bound on the log-partition function

$$\begin{aligned} c(\theta) &= \log \sum_{\mathbf{z} \in \mathscr{Z}} \exp \langle \theta, \mathbf{z} \rangle \\ &= \max_{\mathbf{p} \in \triangle^k} \langle \mathbf{p}, \mathrm{U}\theta \rangle \ \underbrace{-\ \sum_i p_i \log p_i}_{=H[\mathbf{p}]} \\ &= \max_{\mathbf{p} \in \triangle^k} \underbrace{\left(\mathbf{p}^\top \mathrm{U}\right)} \theta \ + \ H[\mathbf{p}] \\ &= \max_{\mathbf{z} \in \mathrm{conv}(\mathscr{Z})} \langle \mathbf{z}, \theta \rangle \ + \ \Omega(\mathbf{z}) \end{aligned}$$

Marginal polytope

Implicitly defined so the two problems are equivalent

VARIATIONAL APPROXIMATION

Change of notation

Set of feasible ASTs

$$\mathbf{z} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \qquad \theta = \begin{bmatrix}\mu\\ \phi\end{bmatrix} \qquad \mathscr{Z} = \{\ \mathbf{z}^{(1)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k)}\ \}$$

Upper bound on the log-partition function

$$c(\theta) = \log \sum_{\mathbf{z}\in\mathscr{Z}} \exp\ \langle \theta, \mathbf{z}\rangle$$

$$= \max_{\mathbf{p}\in\triangle^k} \langle \mathbf{p}, \mathbf{U}\theta\rangle\ \underbrace{-\ \sum_i p_i \log p_i}_{=H[\mathbf{p}]}$$

$$= \max_{\mathbf{p}\in\triangle^k} \underbrace{\left(\mathbf{p}^\top \mathbf{U}\right)}\theta\ +\ H[\mathbf{p}]$$

$$= \max_{\mathbf{z}\in\mathrm{conv}(\mathscr{Z})} \langle \mathbf{z}, \theta\rangle\ +\ \Omega(\mathbf{z})$$

$$\leq \max_{\mathbf{z}\in\mathscr{L}} \langle \mathbf{z}, \theta\rangle\ +\ H(\mathbf{z})$$

Mean regularization

Outer approximation

VARIATIONAL APPROXIMATION

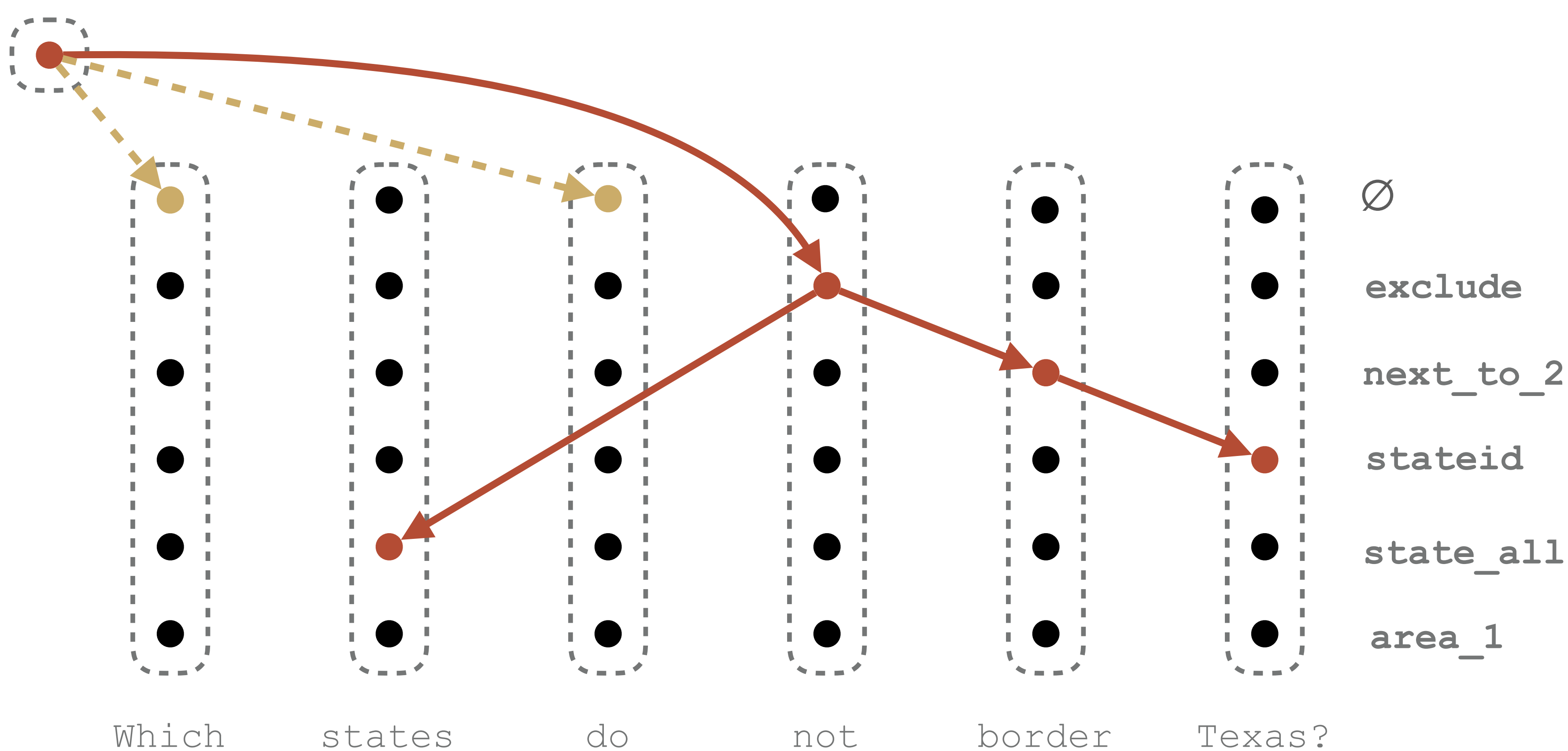
Upper bound on the log-partition function

$$c(\theta) \leq \max_{\mathbf{z}\in\mathscr{L}} \langle \mathbf{z}, \theta \rangle + H(\mathbf{z}) = \widetilde{c}(\theta)$$

We need to choose $\mathscr{L}$ such that the bound is easy to compute.

Note that each feasible solution in $\mathscr{C}$ satisfies the following conditions:

1. Each cluster has exactly one selected vertex
2. Each cluster (except the root) has exactly one incoming arc



VARIATIONAL APPROXIMATION

Upper bound on the log-partition function

$$c(\theta) \leq \max_{\mathbf{z} \in \mathscr{L}} \langle \mathbf{z}, \theta \rangle + H(\mathbf{z}) = \widetilde{c}(\theta)$$

We need to choose $\mathscr{L}$ such that the bound is easy to compute.

Note that each feasible solution in $\mathscr{C}$ satisfies the following conditions:

1. Each cluster has exactly one selected vertex
2. Each cluster (except the root) has exactly one incoming arc

Token-separable negative log-likelihood

Define $\mathscr{L}$ as the convex hull of structures that satisfy (1) and (2),

Then:

$$\ell(\mu, \phi; \mathbf{x}, \mathbf{y}) \leq -\langle \mu, \mathbf{x} \rangle - \langle \phi, \mathbf{y} \rangle + \widetilde{c}(\mu, \phi)$$

is simply a sum of negative log-likelihood losses. For each cluster:

- One NLL over all vertices in the cluster
- One NLL over all incoming arcs in the cluster

WEAKLY-SUPERVISED LEARNING

DATASETS

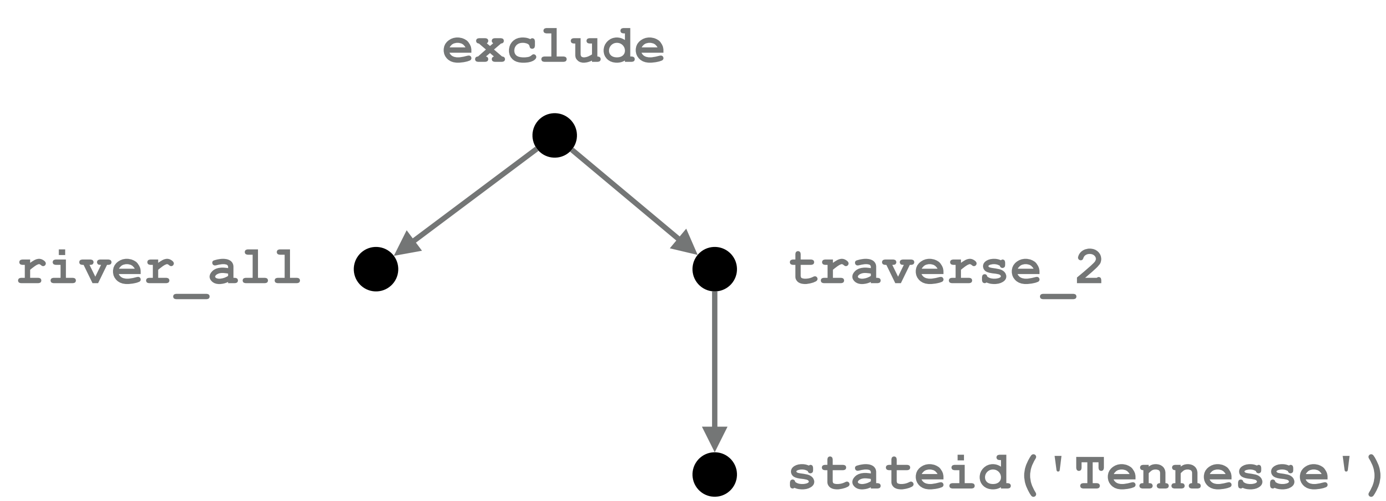
Annotation issue

In most dataset, the entities and predicates are not anchored!

Example

Input: `What rivers do not run through Tennesse?`

Output:



WEAKLY SUPERVISED LOSS

$$\widetilde{\ell}(\mu, \phi; \mathscr{C}^*) = -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} p_{\mu,\phi}(\mathbf{x}, \mathbf{y}) \;= -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\; \langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle - c(\mu, \phi) \;\big)$$

$$= -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\; \langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle \;\big) \quad + \quad c(\mu, \phi)$$

WEAKLY SUPERVISED LOSS

$$\widetilde{\ell}(\mu, \phi; \mathscr{C}^*) = -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} p_{\mu,\phi}(\mathbf{x}, \mathbf{y}) \;= -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\, \langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle - c(\mu, \phi) \,\big)$$

$$= -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\, \langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle \,\big) \quad + \quad c(\mu, \phi)$$

Lower bound on the first term

$$\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\, \langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle \,\big) \;=\; \log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \frac{q(\mathbf{x}, \mathbf{y})}{q(\mathbf{x}, \mathbf{y})} \exp\big(\, \langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle \,\big)$$

Proposal distribution

WEAKLY SUPERVISED LOSS

$$\widetilde{\ell}(\mu, \phi; \mathscr{C}^*) = -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} p_{\mu,\phi}(\mathbf{x}, \mathbf{y}) = -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle - c(\mu, \phi) \big)$$

$$= -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle \big) \quad + \quad c(\mu, \phi)$$

Lower bound on the first term

$$\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle \big) = \log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \frac{q(\mathbf{x}, \mathbf{y})}{q(\mathbf{x}, \mathbf{y})} \exp\big(\langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle \big)$$

$$\geq \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} q(\mathbf{x}, \mathbf{y}) \log \frac{\exp\big(\langle \mu, \mathbf{x} \rangle + \langle \phi, \mathbf{y} \rangle \big)}{q(\mathbf{x}, \mathbf{y})}$$

Jensen's inequality

WEAKLY SUPERVISED LOSS

$$\widetilde{\ell}(\mu, \phi; \mathscr{C}^*) = -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} p_{\mu,\phi}(\mathbf{x}, \mathbf{y}) = -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\ \langle \mu, \mathbf{x}\rangle + \langle \phi, \mathbf{y}\rangle - c(\mu, \phi)\ \big)$$

$$= -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\ \langle \mu, \mathbf{x}\rangle + \langle \phi, \mathbf{y}\rangle\ \big) \quad + \quad c(\mu, \phi)$$

Lower bound on the first term

$$\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \exp\big(\ \langle \mu, \mathbf{x}\rangle + \langle \phi, \mathbf{y}\rangle\ \big) \ = \ \log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} \frac{q(\mathbf{x}, \mathbf{y})}{q(\mathbf{x}, \mathbf{y})} \exp\big(\ \langle \mu, \mathbf{x}\rangle + \langle \phi, \mathbf{y}\rangle\ \big)$$

$$\geq \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}^*} q(\mathbf{x}, \mathbf{y}) \ \log \frac{\exp\big(\ \langle \mu, \mathbf{x}\rangle + \langle \phi, \mathbf{y}\rangle\ \big)}{q(\mathbf{x}, \mathbf{y})}$$

$$= \ \mathbb{E}_q\big[\ \langle \mu, \mathbf{x}\rangle \ + \ \langle \phi, \mathbf{y}\rangle\ \big] \ + \ H[q]$$

As usual:

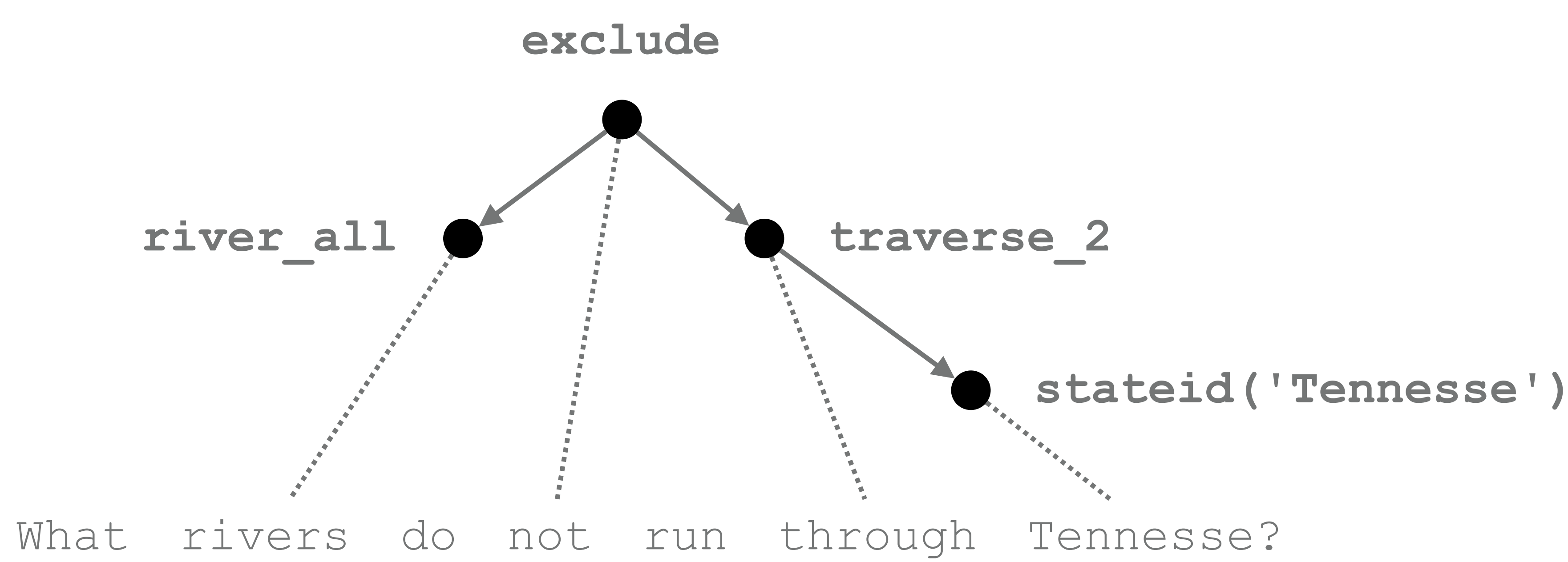
- The bound is tight if q is equal to the posterior distribution, "à la" EM
- We can instead use a proposal that put all the mass on single value, "à la" hard EM

WEAKLY SUPERVISED LOSS

$$\widetilde{\ell}(\mu, \phi; \mathscr{C}*) = -\log \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{C}*} p_{\mu,\phi}(\mathbf{x}, \mathbf{y}) \quad \leq \quad \mathbb{E}_q\left[\,\langle \mu, \mathbf{x}\rangle \;+\; \langle \phi, \mathbf{y}\rangle\,\right] \;+\; H[q] \quad + \quad \widetilde{c}(\mu, \phi)$$

Hard-EM like optimization

- **(E step)** Compute the best alignment between vertices in the AST and words in the sentence
- **(M step)** One gradient step on the neural network parameters



NP-hardness

The E step is a NP-hard problem => approximate solver based on constraint relaxation
+ dynamic programming

EXPERIMENTAL RESULTS

DATASETS

SCAN: Simplified version of the CommAI Navigation tasks [Lake & Baroni, 2018]

- Input : command
- Output : action sequence

```
jump ⇒ JUMP
jump left ⇒ LTURN JUMP
jump around right ⇒ RTURN JUMP RTURN JUMP RTURN JUMP RTURN JUMP
turn left twice ⇒ LTURN LTURN
jump thrice ⇒ JUMP JUMP JUMP
jump opposite left and walk thrice ⇒ LTURN LTURN JUMP WALK WALK WALK
jump opposite left after walk around left
⇒ LTURN WALK LTURN WALK LTURN WALK LTURN WALK LTURN LTURN JUMP
```

SCAN-SP [Herzig & Berant, 2021]

Variant of scan where outputs are reformulated as functional programs

```
run around left twice and jump left
⇒ i_and ( i_twice ( i_run ( i_left , i_around ) ) , i_jump ( i_left ) )
```

DATASETS

SCAN : IID

Random split of the data

DATASETS

SCAN : IID

Random split of the data

SCAN : Right

- The term "right" is never seen without a manner adverbs (around, opposite) during training
- The model must learn to generalize to the simplest usage of right
  (as seen during training for "left")

Train	Test
`jump left`	`jump right`
`turn left`	`turn right`
`jump around left`	`...`
`jump around right`	
`turn opposite right`	
`turn around left`	
`...`	

DATASETS

SCAN : IID

Random split of the data

SCAN : Right

- The term "right" is never seen without a manner adverbs (around, opposite) during training
- The model must learn to generalize to the simplest usage of right
(as seen during training for "left")

SCAN : Around right

- Test test set contains all exemple with "around right"
- The train set contains all other examples

Train	Test
`jump left`	`jump around right`
`jump right`	`turn around right`
`jump around left`	`...`
`jump opposite right`	
`turn opposite right`	
`turn around left`	
`...`	

DATASETS

SCAN : IID

Random split of the data

SCAN : Right

- The term "right" is never seen without a manner adverbs (around, opposite) during training
- The model must learn to generalize to the simplest usage of right
  (as seen during training for "left")

SCAN : Around right

- Test test set contains all exemple with "around right"
- The train set contains all other examples

	train	dev	test
IID	13 383	3 345	4 182
Right	12 180	3 045	4 476
ARight	12 180	3 045	4 476

DATASETS

GeoQuery

- Input: question related to USA geography
- Output: query that can be executed against a database

```
what state has the largest city? ⇒ answer(state(loc_1(largest(city(all)))))
how many square kilometers in the us? ⇒ answer(area_1(countryid('usa')))
```

DATASETS

GeoQuery

- Input: question related to USA geography
- Output: query that can be executed against a database

```
what state has the largest city? ⇒ answer(state(loc_1(largest(city(all)))))
how many square kilometers in the us? ⇒ answer(area_1(countryid('usa')))
```

SCAN : IID

Random split of the data

DATASETS

GeoQuery

- Input: question related to USA geography
- Output: query that can be executed against a database

```
what state has the largest city? ⇒ answer(state(loc_1(largest(city(all)))))
how many square kilometers in the us? ⇒ answer(area_1(countryid('usa')))
```

SCAN : IID

Random split of the data

SCAN : Template

All sentences that shares the same semantic template are used only for training or only for testing.

```
name the rivers in arkansas
name all the rivers in colorado
name all the rivers in colorado
rivers in new york ?
what are all the rivers in texas ?
...
```

DATASETS

GeoQuery

- Input: question related to USA geography
- Output: query that can be executed against a database

```
what state has the largest city? ⇒ answer(state(loc_1(largest(city(all)))))
how many square kilometers in the us? ⇒ answer(area_1(countryid('usa')))
```

SCAN : IID

Random split of the data

SCAN : Template

All sentences that shares the same semantic template are used only for training or only for testing.

SCAN : Length

Test sentences are (in average) longer than train sentences

Train

- sentence length: min=4 / max=13 / mean=7.5
- program length: min=1 / max=4 / mean=3.1

Test

- sentence length: min=7 / max=18 / mean=10.5
- program length: min=2 / max=9 / mean=5.2

DATASETS

GeoQuery

- Input: question related to USA geography
- Output: query that can be executed against a database

```
what state has the largest city? ⇒ answer(state(loc_1(largest(city(all)))))
how many square kilometers in the us? ⇒ answer(area_1(countryid('usa')))
```

SCAN : IID

Random split of the data

SCAN : Template

All sentences that shares the same semantic template are used only for training or only for testing.

SCAN : Length

Test sentences are (in average) longer than train sentences

	train	dev	test
IID	540	60	280
Template	544	60	276
Length	540	60	280

DATASETS

Clevr

- Input: question related to objects in a picture
- Output: query that can be executed against a database

```
Are there any shiny objects that have the same color as the matte block?

⇒ exist(filter(metal,relate_att_eq(color,filter(rubber,cube,scene()))))
```

DATASETS

Clevr

- Input: question related to objects in a picture
- Output: query that can be executed against a database

```
Are there any shiny objects that have the same color as the matte block?
⇒ exist(filter(metal,relate_att_eq(color,filter(rubber,cube,scene()))))
```

SCAN : IID

Random split of the data

DATASETS

Clevr

- Input: question related to objects in a picture
- Output: query that can be executed against a database

```
Are there any shiny objects that have the same color as the matte block?

⇒ exist(filter(metal,relate_att_eq(color,filter(rubber,cube,scene()))))
```

SCAN : IID

Random split of the data

SCAN : Closure

- Questions in Clevr are generated from 80 templates
- Questions in Closure are generated from 7 new templates

DATASETS

Clevr

- Input: question related to objects in a picture
- Output: query that can be executed against a database

```
Are there any shiny objects that have the same color as the matte block?

⇒ exist(filter(metal,relate_att_eq(color,filter(rubber,cube,scene()))))
```

SCAN : IID

Random split of the data

SCAN : Closure

- Questions in Clevr are generated from 80 templates
- Questions in Closure are generated from 7 new templates

	train	dev	test
IID	694 689	5 000	149 991
Closure	694 689	5 000	25 200

EXPERIMENTAL RESULTS

	Scan			GeoQuery			Clevr	
	Iid	Right	ARight	Iid	Template	Length	Iid	Closure
Baselines (denotation accuracy only)								
Seq2Seq	99.9	11.6	0	78.5	46.0	24.3	**100**	59.5
+ ELMo	**100**	54.9	41.6	79.3	50.0	25.7	**100**	64.2
BERT2Seq	**100**	77.7	95.3	81.1	49.6	26.1	**100**	56.4
GRAMMAR	**100**	0.0	4.2	72.1	54.0	24.6	**100**	51.3
BART	**100**	50.5	**100**	87.1	67.0	19.3	**100**	51.5
SpanBasedSP	**100**	**100**	**100**	86.1	82.2	63.6	96.7	98.8

All baselines are from [Herzig & Berant, 2021]

EXPERIMENTAL RESULTS

	Scan			GeoQuery			Clevr	
	Iid	Right	ARight	Iid	Template	Length	Iid	Closure
Baselines (denotation accuracy only)								
Seq2Seq	99.9	11.6	0	78.5	46.0	24.3	**100**	59.5
+ ELMo	**100**	54.9	41.6	79.3	50.0	25.7	**100**	64.2
BERT2Seq	**100**	77.7	95.3	81.1	49.6	26.1	**100**	56.4
GRAMMAR	**100**	0.0	4.2	72.1	54.0	24.6	**100**	51.3
BART	**100**	50.5	**100**	87.1	67.0	19.3	**100**	51.5
SpanBasedSP	**100**	**100**	**100**	86.1	82.2	63.6	96.7	98.8
Our approach								
Denotation accuracy	**100**	**100**	**100**	**92.9**	**89.9**	**74.9**	**100**	**99.6**
↳ Corrected executor				91.8	88.7	74.5		
Exact match	**100**	**100**	**100**	90.7	86.2	69.3	**100**	**99.6**
↳ w/o CPLEX heuristic	**100**	**100**	**100**	90.0	83.0	67.5	**100**	98.0

Neural network

BERT-base + BiLSTM + Biaffine (details in the appendix of the paper)

TOKEN-SEPARABLE LOSS FUNCTIONS

Related publication

On the inconsistency of separable losses for structured prediction

Caio Corro

EACL 2023

LOSS FUNCTIONS AND BAYES CONSISTENCY

Motivations

We approximate the log-partition function in the loss,

how does this impact the solution of the training problem?

LOSS FUNCTIONS AND BAYES CONSISTENCY

Motivations

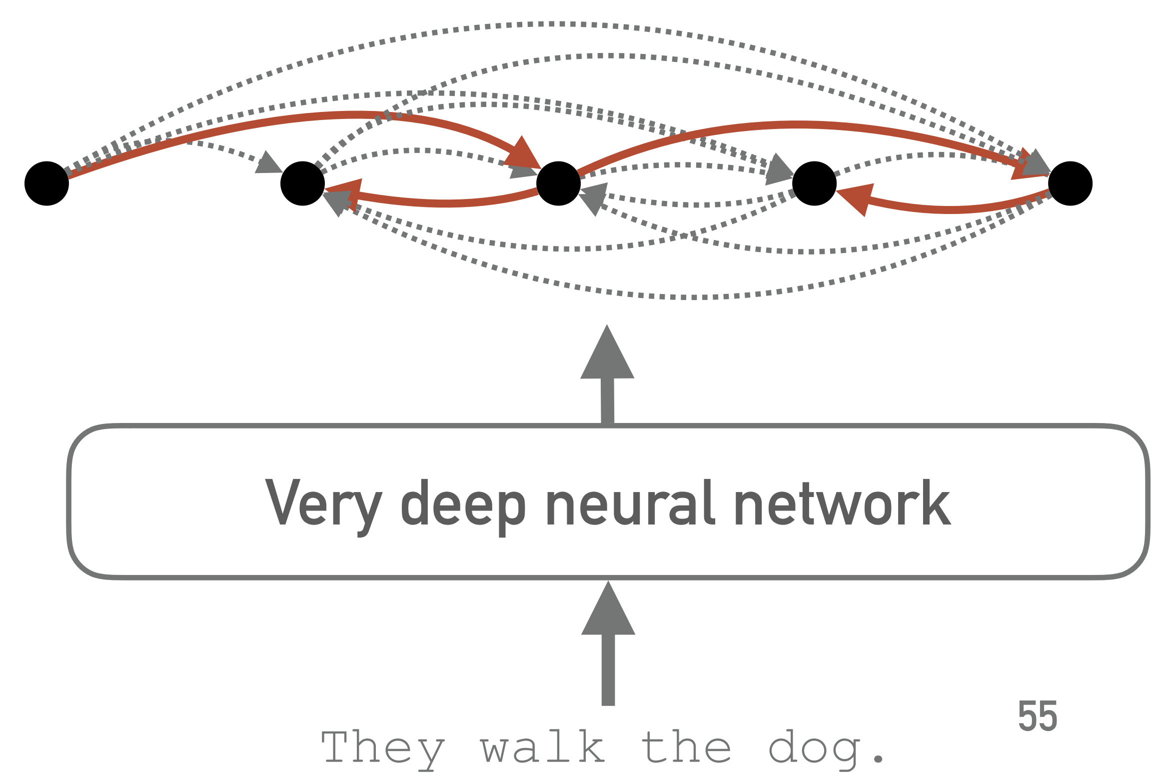
We approximate the log-partition function in the loss,

how does this impact the solution of the training problem?

Simpler example: syntactic dependency parsing

- Compute the maximum spanning arborescence : $\mathcal{O}(n^2)$ **[Tarjan, 1977]**
- Summing over all arborescences : $\mathcal{O}(n^3)$ (via the matrix tree theorem, MTT)
  - Numerically instable (matrix inversion)
  - Not very fast on GPU compared to simpler losses
  - Non-trivial to implement

[Koo et al., 2007] [McDonald & Satta, 2007] [Smith & Smith, 2007]



LOSS FUNCTIONS AND BAYES CONSISTENCY

Motivations

We approximate the log-partition function in the loss,

how does this impact the solution of the training problem?

Simpler example: syntactic dependency parsing

- Compute the maximum spanning arborescence : $\mathcal{O}(n^2)$ [Tarjan, 1977]
- Summing over all arborescences : $\mathcal{O}(n^3)$ (via the matrix tree theorem, MTT)
  - Numerically instable (matrix inversion)
  - Not very fast on GPU compared to simpler losses
  - Non-trivial to implement

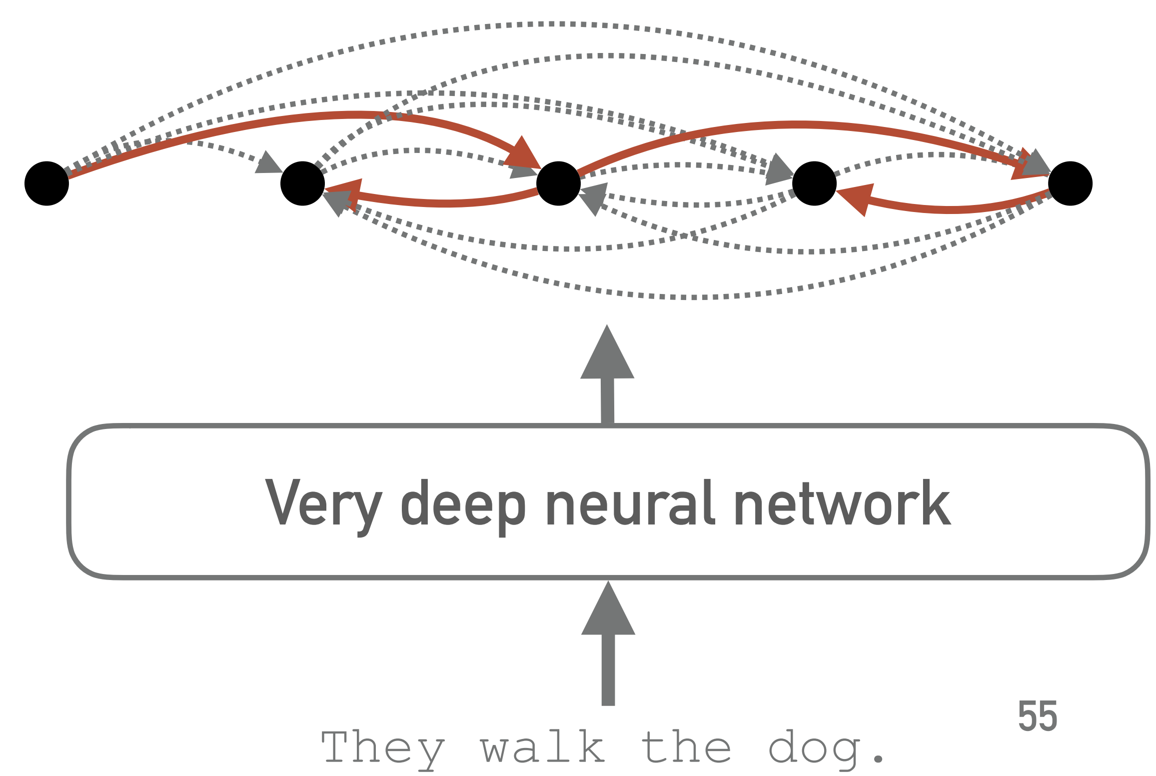
[Koo et al., 2007] [McDonald & Satta, 2007] [Smith & Smith, 2007]

Head selection loss [Zhang et al., 2017]

As each word has exactly one head

=> one multi-class classification loss per word

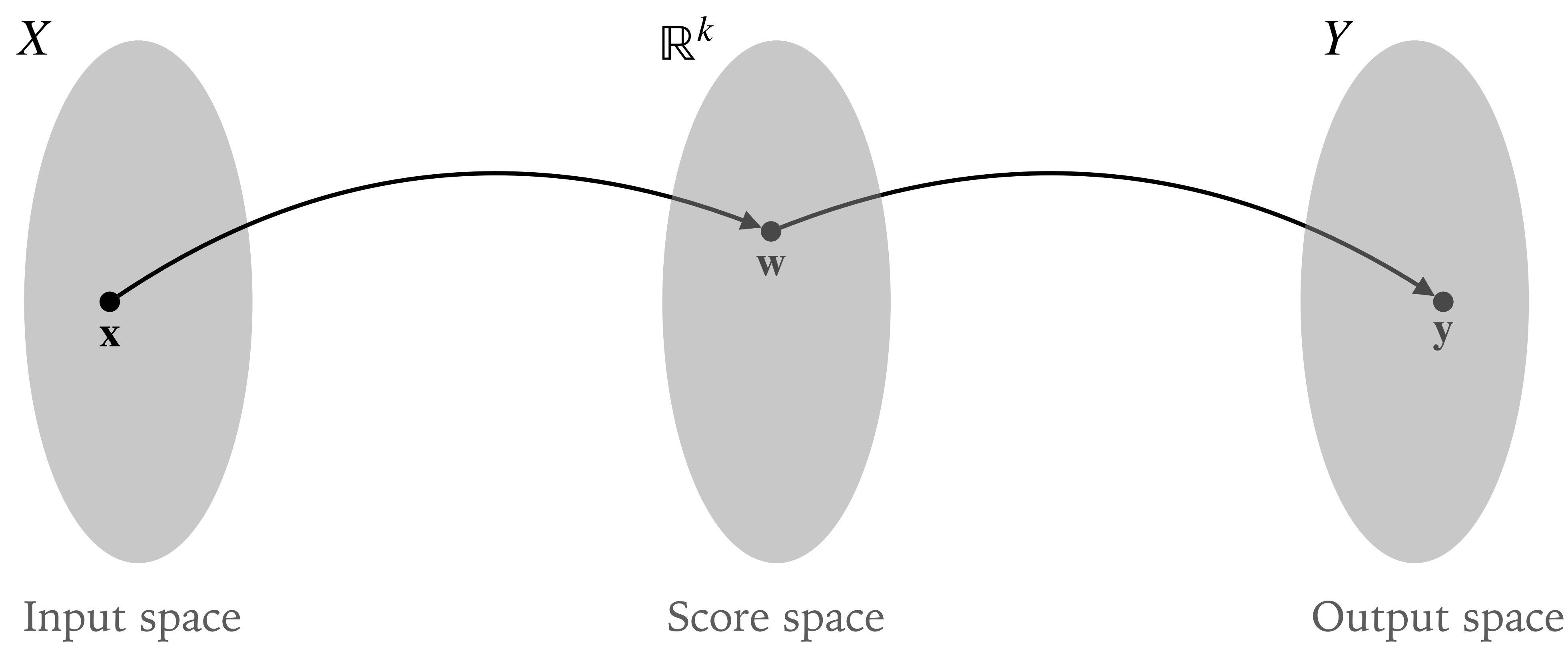
(equivalent to log-partition approximation)



MULTICLASS CLASSIFICATION

Notations

- k : number of classes
- X : input space
- Y : output space, set of one-hot vectors of dimension k
- $f : X \to \mathbb{R}^k$: scoring function
- $\hat{\mathbf{y}} : \mathbb{R}^k \to Y$: prediction function, $\hat{\mathbf{y}}(\mathbf{w}) = \arg\max_{\mathbf{y} \in Y} \langle \mathbf{w}, \mathbf{y} \rangle$



BAYES RISK MINIMIZATION

0-1 loss function

Returns 1 if the output will be incorrect for a given score vector

$$\ell : \mathbb{R}^k \times Y \to \mathbb{R}_+ \qquad \ell(\mathbf{w}, \mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \in \arg\max_{\mathbf{y}' \in Y} \langle \mathbf{y}', \mathbf{w} \rangle, \\ 1 & \text{otherwise.} \end{cases}$$

BAYES RISK MINIMIZATION

0-1 loss function

Returns 1 if the output will be incorrect for a given score vector

$$\ell : \mathbb{R}^k \times Y \to \mathbb{R}_+ \qquad \ell(\mathbf{w}, \mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \in \arg\max_{\mathbf{y}' \in Y} \langle \mathbf{y}', \mathbf{w} \rangle, \\ 1 & \text{otherwise.} \end{cases}$$

Optimal Bayes risk

Given a set of scoring function F, what is minimum average number of error we can obtain?

$$\underbrace{r^*}_{\text{Optimal Bayes risk}} = \inf_{\mathbf{f} \in F} r(f) = \inf_{f \in F} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \ell(f(\mathbf{x}), \mathbf{y})\ ] = \mathbb{E}_{\mathbf{x}}[\ 1 - \max_{\mathbf{y} \in Y} p(\mathbf{y} \mid \mathbf{x})\ ]$$

BAYES RISK MINIMIZATION

0-1 loss function

Returns 1 if the output will be incorrect for a given score vector

$$\ell : \mathbb{R}^k \times Y \to \mathbb{R}_+ \qquad \ell(\mathbf{w}, \mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \in \arg\max_{\mathbf{y}' \in Y} \langle \mathbf{y}', \mathbf{w} \rangle, \\ 1 & \text{otherwise.} \end{cases}$$

Optimal Bayes risk

Given a set of scoring function F, what is minimum average number of error we can obtain?

$$r^* = \inf_{\mathbf{f} \in F} \underbrace{r(f)}_{\text{Bayes risk of f}} = \inf_{f \in F} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \ell(f(\mathbf{x}), \mathbf{y})\ ] = \mathbb{E}_{\mathbf{x}}[\ 1 - \max_{\mathbf{y} \in Y} p(\mathbf{y} \,|\, \mathbf{x})\ ]$$

BAYES RISK MINIMIZATION

0-1 loss function

Returns 1 if the output will be incorrect for a given score vector

$$\ell : \mathbb{R}^k \times Y \to \mathbb{R}_+ \qquad \ell(\mathbf{w}, \mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \in \arg\max_{\mathbf{y}' \in Y} \langle \mathbf{y}', \mathbf{w} \rangle, \\ 1 & \text{otherwise.} \end{cases}$$

Optimal Bayes risk

Given a set of scoring function F, what is minimum average number of error we can obtain?

$$r^* = \inf_{\mathbf{f} \in F} r(f) = \inf_{f \in F} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \ell(f(\mathbf{x}), \mathbf{y})\ ] = \mathbb{E}_{\mathbf{x}}[\ 1 - \max_{\mathbf{y} \in Y} p(\mathbf{y} \,|\, \mathbf{x})\ ]$$

Training objective!

BAYES RISK MINIMIZATION

0-1 loss function

Returns 1 if the output will be incorrect for a given score vector

$$\ell : \mathbb{R}^k \times Y \to \mathbb{R}_+ \qquad \ell(\mathbf{w}, \mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \in \arg\max_{\mathbf{y}' \in Y} \langle \mathbf{y}', \mathbf{w} \rangle, \\ 1 & \text{otherwise.} \end{cases}$$

Optimal Bayes risk

Given a set of scoring function F, what is minimum average number of error we can obtain?

$$r^* = \inf_{\mathbf{f} \in F} r(f) = \inf_{f \in F} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \ell(f(\mathbf{x}), \mathbf{y})\ ] = \mathbb{E}_{\mathbf{x}}[\ 1 - \max_{\mathbf{y} \in Y} p(\mathbf{y} \,|\, \mathbf{x})\ ]$$

Bayes risk when we predict the most probable output for each input

BAYES RISK MINIMIZATION

0-1 loss function

Returns 1 if the output will be incorrect for a given score vector

$$\ell : \mathbb{R}^k \times Y \to \mathbb{R}_+ \qquad \ell(\mathbf{w}, \mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \in \arg\max_{\mathbf{y}' \in Y} \langle \mathbf{y}', \mathbf{w} \rangle, \\ 1 & \text{otherwise.} \end{cases}$$

Optimal Bayes risk

Given a set of scoring function F, what is minimum average number of error we can obtain?

$$r^* \;=\; \inf_{\mathbf{f} \in F} \; r(f) \;=\; \inf_{f \in F} \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \ell(f(\mathbf{x}), \mathbf{y}) \;] \;=\; \mathbb{E}_{\mathbf{x}}[\; 1 - \max_{\mathbf{y} \in Y} p(\mathbf{y} \,|\, \mathbf{x}) \;]$$

Bayes risk minimization

- The 0-1 loss function is not convex in $\mathbf{w}$
- The derivatives of the objective are null a.e.
- The problem is know to be intractable even in simple cases

SURROGATE LOSSES

Motivations

We can not use the 0-1 loss ℓ for training, therefore we want to use a surrogate loss $\widetilde{\ell}$,

are solutions of the surrogate training problem <u>optimal Bayes classifiers</u>?

SURROGATE LOSSES

Motivations

We can not use the 0-1 loss ℓ for training, therefore we want to use a surrogate loss $\widetilde{\ell}$,

are solutions of the surrogate training problem <u>optimal Bayes classifiers</u>?

Surrogate risk

Given a set of scoring function F, what is minimum average number of error we can obtain?

$$\widetilde{r}^* = \inf_{\mathbf{f} \in F} \widetilde{r}(f) = \inf_{f \in F} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \widetilde{\ell}\,(f(\mathbf{x}), \mathbf{y})\ ]$$

SURROGATE LOSSES

Motivations

We can not use the 0-1 loss ℓ for training, therefore we want to use a surrogate loss $\widetilde{\ell}$,

are solutions of the surrogate training problem <u>optimal Bayes classifiers</u>?

Surrogate risk

Given a set of scoring function F, what is minimum average number of error we can obtain?

$$\widetilde{r}^* \quad = \quad \inf_{\mathbf{f} \in F} \ \widetilde{r}(f) \quad = \quad \inf_{f \in F} \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \widetilde{\ell}(f(\mathbf{x}), \mathbf{y})\ ]$$

Bayes consistency

A surrogate loss $\widetilde{\ell}$ is said to be Bayes consistent / Fisher consistent / classification calibrated if:

$$f^* \in \arg\min_{f \in F} \ \widetilde{r}(f) \quad \Longrightarrow \quad r(f^*) = r^*$$

POINTWISE CONSISTENCY

Standard assumptions

- F is the set of all measurable mappings
- Infinite number of training datapoints (i.e. expectation over the "true" data distribution)

Pointwise setting

- Choose a datapoint $\mathbf{x} \in X$ such that $p(\mathbf{x}) > 0$
- Redefine the Bayes and surrogate risks as expectation over the conditional distribution $p(\mathbf{y} \mid \mathbf{x})$
- Minimize over the score vector $\mathbf{w} \in \mathbb{R}^k$ instead of over function set F, where $\mathbf{w} = f(\mathbf{x})$

$$r^* = \inf_{\mathbf{w} \in \mathbb{R}^k} r(\mathbf{w}) = \inf_{\mathbf{w} \in \mathbb{R}^k} \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\ \ell(\mathbf{w}, \mathbf{y})\ ] = 1 - \max_{\mathbf{y} \in Y} p(\mathbf{y} \mid \mathbf{x})$$

$$\widetilde{r}^* = \inf_{\mathbf{w} \in \mathbb{R}^k} \widetilde{r}(\mathbf{w}) = \inf_{\mathbf{w} \in \mathbb{R}^k} \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\ \widetilde{\ell}(\mathbf{w}, \mathbf{y})\ ]$$

NEGATIVE LOG-LIKELIHOOD

Negative log-likelihood loss

$$\widetilde{\ell}(\mathbf{w}, \mathbf{y}) \;=\; -\langle \mathbf{w}, \mathbf{y} \rangle \;+\; \log \sum_{\mathbf{y}' \in Y} \exp\langle \mathbf{w}, \mathbf{y}' \rangle \;=\; -\langle \mathbf{w}, \mathbf{y} \rangle \;+\; c(\mathbf{w})$$

NEGATIVE LOG-LIKELIHOOD

Negative log-likelihood loss

$$\widetilde{\ell}(\mathbf{w}, \mathbf{y}) \;=\; -\langle \mathbf{w}, \mathbf{y} \rangle \;+\; \log \sum_{\mathbf{y}' \in Y} \exp\langle \mathbf{w}, \mathbf{y}' \rangle \;=\; -\langle \mathbf{w}, \mathbf{y} \rangle \;+\; c(\mathbf{w})$$

Surrogate risk

$$\begin{aligned} \inf_{\mathbf{w} \in \mathbb{R}^k} \; \widetilde{r}(\mathbf{w}) \;&=\; \inf_{\mathbf{w} \in \mathbb{R}^k} \; \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\; \widetilde{\ell}(\mathbf{w}, \mathbf{y}) \;] \\ &=\; \inf_{\mathbf{w} \in \mathbb{R}^k} \; \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\, -\langle \mathbf{w}, \mathbf{y} \rangle \;+\; c(\mathbf{w}) \;] \\ &=\; \inf_{\mathbf{w} \in \mathbb{R}^k} \; -\langle \mathbf{w},\, E_{\mathbf{y}|\mathbf{x}}[\mathbf{y}] \;\rangle \;+\; c(\mathbf{w}) \end{aligned}$$

NEGATIVE LOG-LIKELIHOOD

Negative log-likelihood loss

$$\widetilde{\ell}(\mathbf{w}, \mathbf{y}) \;=\; -\langle \mathbf{w}, \mathbf{y} \rangle \;+\; \log \sum_{\mathbf{y}' \in Y} \exp\langle \mathbf{w}, \mathbf{y}' \rangle \;=\; -\langle \mathbf{w}, \mathbf{y} \rangle \;+\; c(\mathbf{w})$$

Surrogate risk

$$\begin{aligned}
\inf_{\mathbf{w} \in \mathbb{R}^k} \widetilde{r}(\mathbf{w}) &= \inf_{\mathbf{w} \in \mathbb{R}^k} \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\, \widetilde{\ell}(\mathbf{w}, \mathbf{y}) \,] \\
&= \inf_{\mathbf{w} \in \mathbb{R}^k} \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\, -\langle \mathbf{w}, \mathbf{y} \rangle \;+\; c(\mathbf{w}) \,] \\
&= \inf_{\mathbf{w} \in \mathbb{R}^k} -\langle \mathbf{w}, E_{\mathbf{y}|\mathbf{x}}[\mathbf{y}] \,\rangle \;+\; c(\mathbf{w})
\end{aligned}$$

Optimality conditions

Let:
- $\widehat{\mathbf{w}}$ be a minimizer of the problem above
- $\mathbf{y}^{(i)}$ the one-hot vector for which $\mathbf{y}_i^{(i)} = 1$

By first order optimality conditions:

$$\frac{\partial}{\partial \widehat{w}_i} \Big(-\langle \widehat{\mathbf{w}}, E_{\mathbf{y}|\mathbf{x}}[\mathbf{y}] \,\rangle \;+\; c(\widehat{\mathbf{w}}) \Big) = 0$$

NEGATIVE LOG-LIKELIHOOD

Negative log-likelihood loss

$$\widetilde{\ell}(\mathbf{w},\mathbf{y}) \;=\; -\langle \mathbf{w},\mathbf{y}\rangle \;+\; \log \sum_{\mathbf{y}'\in Y} \exp\langle \mathbf{w},\mathbf{y}'\rangle \;=\; -\langle \mathbf{w},\mathbf{y}\rangle \;+\; c(\mathbf{w})$$

Surrogate risk

$$\begin{aligned}
\inf_{\mathbf{w}\in\mathbb{R}^k} \; \widetilde{r}(\mathbf{w}) &= \inf_{\mathbf{w}\in\mathbb{R}^k} \; \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\; \widetilde{\ell}(\mathbf{w},\mathbf{y}) \;] \\
&= \inf_{\mathbf{w}\in\mathbb{R}^k} \; \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\, -\langle \mathbf{w},\mathbf{y}\rangle \;+\; c(\mathbf{w}) \;] \\
&= \inf_{\mathbf{w}\in\mathbb{R}^k} \; -\langle \mathbf{w},\, E_{\mathbf{y}|\mathbf{x}}[\mathbf{y}] \;\rangle \;+\; c(\mathbf{w})
\end{aligned}$$

Optimality conditions

Let:
- $\widehat{\mathbf{w}}$ be a minimizer of the problem above
- $\mathbf{y}^{(i)}$ the one-hot vector for which $\mathbf{y}_i^{(i)} = 1$

By first order optimality conditions:

$$\frac{\partial}{\partial \widehat{w}_i}\Big(-\langle\, \widehat{\mathbf{w}}\,,\, E_{\mathbf{y}|\mathbf{x}}[\mathbf{y}] \;\rangle \;+\; c(\,\widehat{\mathbf{w}}\,)\Big) = 0 \qquad \Longrightarrow \qquad \frac{\exp \widehat{w}_i}{\sum_j \exp \widehat{w}_j} = p(\mathbf{y}^{(i)}\,|\,\mathbf{x})$$

Bayes consistent!

EXAMPLE

Optimality conditions

$$\frac{\partial}{\partial \widehat{w}_i}\left(-\langle \widehat{\mathbf{w}}, E_{\mathbf{y}|\mathbf{x}}[\mathbf{y}] \rangle + c(\widehat{\mathbf{w}})\right) = 0 \qquad \Longrightarrow \qquad \frac{\exp \widehat{w}_i}{\sum_j \exp \widehat{w}_j} = p(\mathbf{y}^{(i)} \,|\, \mathbf{x})$$

$$\Longrightarrow \qquad \widehat{w}_i = \log p(\mathbf{y}^{(i)} \,|\, \mathbf{x})$$

Example

$$p(\mathbf{y}^{(1)} \,|\, \mathbf{x}) = 0.7$$

$$p(\mathbf{y}^{(2)} \,|\, \mathbf{x}) = 0.1$$

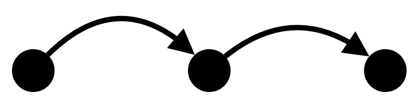
$$p(\mathbf{y}^{(3)} \,|\, \mathbf{x}) = 0.2$$

	$\widehat{\mathbf{w}}$
1	$\log 0.7$
2	$\log 0.1$
3	$\log 0.2$

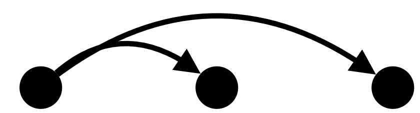
DEPENDENCY PARSING

Distribution over dependency trees

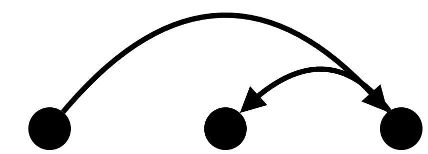
- Sentence length: 2
- No single root constraint



$p(\mathbf{a} \mid \mathbf{x}) = 0.4$



$p(\mathbf{b} \mid \mathbf{x}) = 0.3$

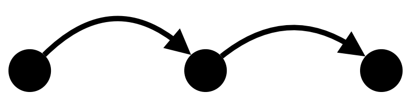


$p(\mathbf{c} \mid \mathbf{x}) = 0.3$

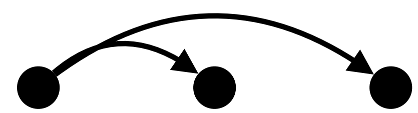
DEPENDENCY PARSING

Distribution over dependency trees

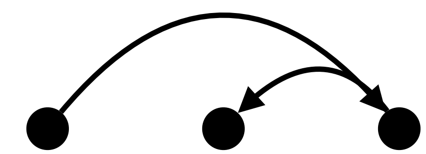
- Sentence length: 2
- No single root constraint



$p(\mathbf{a} \mid \mathbf{x}) = 0.4$



$p(\mathbf{b} \mid \mathbf{x}) = 0.3$



$p(\mathbf{c} \mid \mathbf{x}) = 0.3$

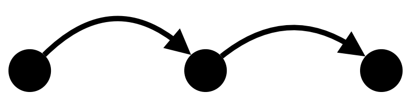
Arc factored scoring function

$\mathbf{w}(\mathbf{a}) = w_{0\to1} + w_{1\to2}$ $\mathbf{w}(\mathbf{b}) = w_{0\to1} + w_{0\to2}$ $\mathbf{w}(\mathbf{c}) = w_{0\to2} + w_{2\to1}$

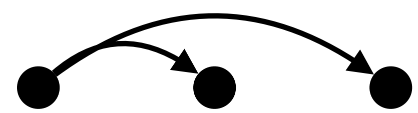
DEPENDENCY PARSING

Distribution over dependency trees

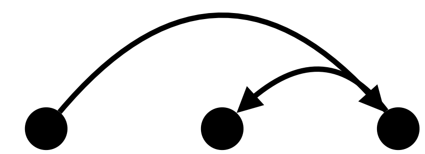
- Sentence length: 2
- No single root constraint



$p(\mathbf{a} \,|\, \mathbf{x}) = 0.4$



$p(\mathbf{b} \,|\, \mathbf{x}) = 0.3$



$p(\mathbf{c} \,|\, \mathbf{x}) = 0.3$

Arc factored scoring function

$$\mathbf{w}(\mathbf{a}) = w_{0\to1} + w_{1\to2} \qquad \mathbf{w}(\mathbf{b}) = w_{0\to1} + w_{0\to2} \qquad \mathbf{w}(\mathbf{c}) = w_{0\to2} + w_{2\to1}$$

Optimality conditions

$$\widehat{\mathbf{w}}(\mathbf{a}) = \log p(\mathbf{a} \,|\, \mathbf{x})$$

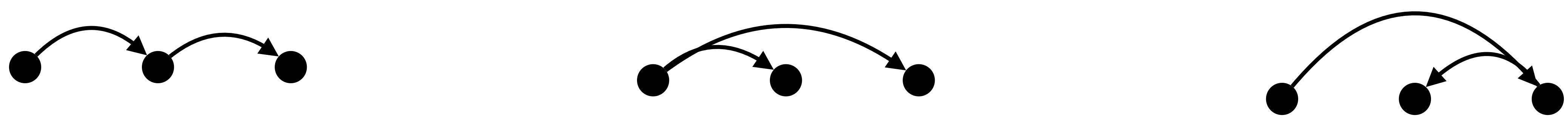
$$\widehat{\mathbf{w}}(\mathbf{b}) = \log p(\mathbf{b} \,|\, \mathbf{x})$$

$$\widehat{\mathbf{w}}(\mathbf{c}) = \log p(\mathbf{c} \,|\, \mathbf{x})$$

DEPENDENCY PARSING

Distribution over dependency trees

- Sentence length: 2
- No single root constraint



$p(\mathbf{a} \mid \mathbf{x}) = 0.4$ $\quad$ $p(\mathbf{b} \mid \mathbf{x}) = 0.3$ $\quad$ $p(\mathbf{c} \mid \mathbf{x}) = 0.3$

Arc factored scoring function

$\mathbf{w}(\mathbf{a}) = w_{0\to1} + w_{1\to2}$ $\quad$ $\mathbf{w}(\mathbf{b}) = w_{0\to1} + w_{0\to2}$ $\quad$ $\mathbf{w}(\mathbf{c}) = w_{0\to2} + w_{2\to1}$

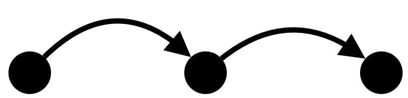
Optimality conditions

$$\begin{aligned} \widehat{\mathbf{w}}(\mathbf{a}) &= \log p(\mathbf{a} \mid \mathbf{x}) \\ \widehat{\mathbf{w}}(\mathbf{b}) &= \log p(\mathbf{b} \mid \mathbf{x}) \\ \widehat{\mathbf{w}}(\mathbf{c}) &= \log p(\mathbf{c} \mid \mathbf{x}) \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \widehat{w}_{0\to1} + \widehat{w}_{1\to2} &= \log p(\mathbf{a} \mid \mathbf{x}) \\ \widehat{w}_{0\to1} + \widehat{w}_{0\to2} &= \log p(\mathbf{b} \mid \mathbf{x}) \\ \widehat{w}_{0\to2} + \widehat{w}_{2\to1} &= \log p(\mathbf{c} \mid \mathbf{x}) \end{aligned}$$

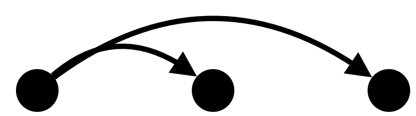
DEPENDENCY PARSING

Distribution over dependency trees

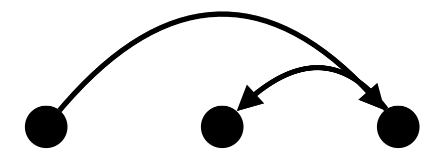
- Sentence length: 2
- Single root



$p(\mathbf{a}\,|\,\mathbf{x}) = 0.4$



$p(\mathbf{b}\,|\,\mathbf{x}) = 0.3$



$p(\mathbf{c}\,|\,\mathbf{x}) = 0.3$

Optimality conditions

$$\widehat{w}_{0\to1} + \widehat{w}_{1\to2} = \log p(\mathbf{a}\,|\,\mathbf{x})$$

$$\widehat{w}_{0\to1} + \widehat{w}_{0\to2} = \log p(\mathbf{b}\,|\,\mathbf{x})$$

$$\widehat{w}_{0\to2} + \widehat{w}_{2\to1} = \log p(\mathbf{c}\,|\,\mathbf{x})$$

Modifier index

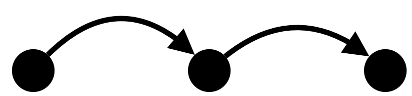
$\widehat{\mathbf{w}}$	0	1	2
0	/	0	$\log 0.3$
1	/	/	$\log 0.4$
2	/	0	/

Head index

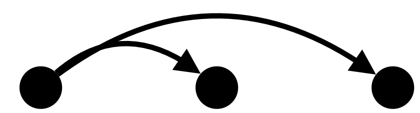
TOKEN-SEPARABLE LOSS FUNCTIONS

Distribution over dependency trees

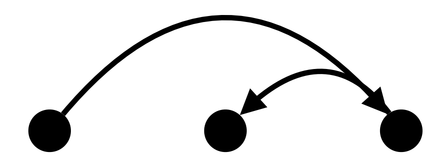
- Sentence length: 2
- Single root



$p(\mathbf{a} \,|\, \mathbf{x}) = 0.4$



$p(\mathbf{b} \,|\, \mathbf{x}) = 0.3$

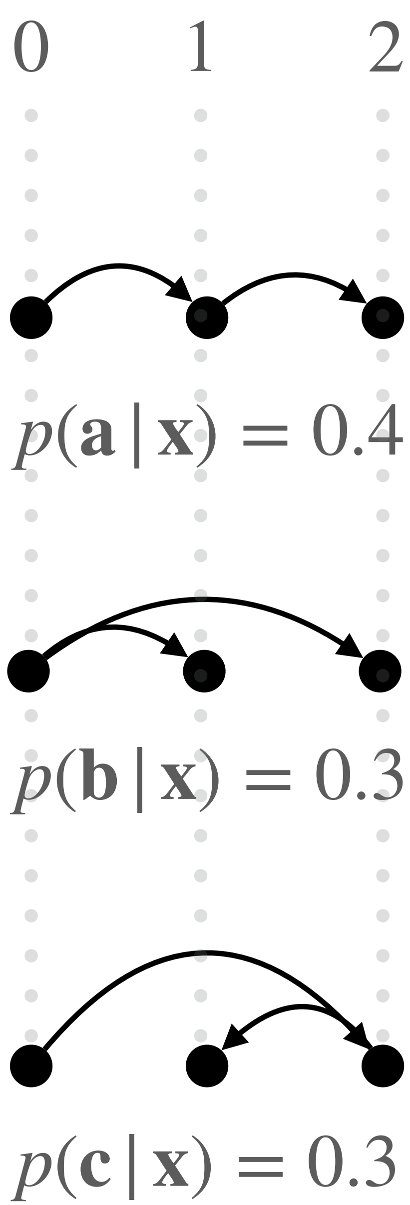


$p(\mathbf{c} \,|\, \mathbf{x}) = 0.3$

Main idea

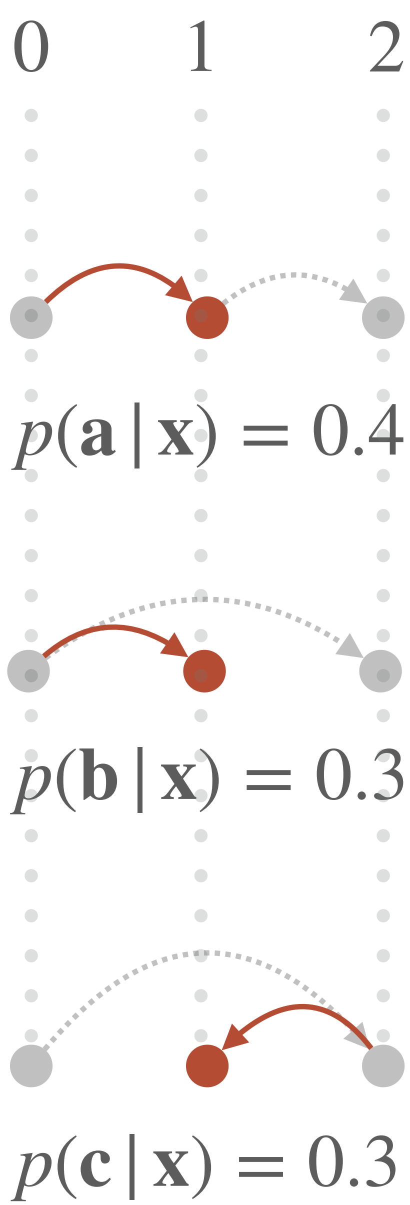
As each word has exactly one head, instead of minimizing the NLL over the dependency tree distribution, **we can minimize one multiclass classification NLL per word**

TOKEN-SEPARABLE LOSS FUNCTIONS



$\hat{\mathbf{w}}$	0	1	2
0	/	?	?
1	/	/	?
2	/	?	/

TOKEN-SEPARABLE LOSS FUNCTIONS

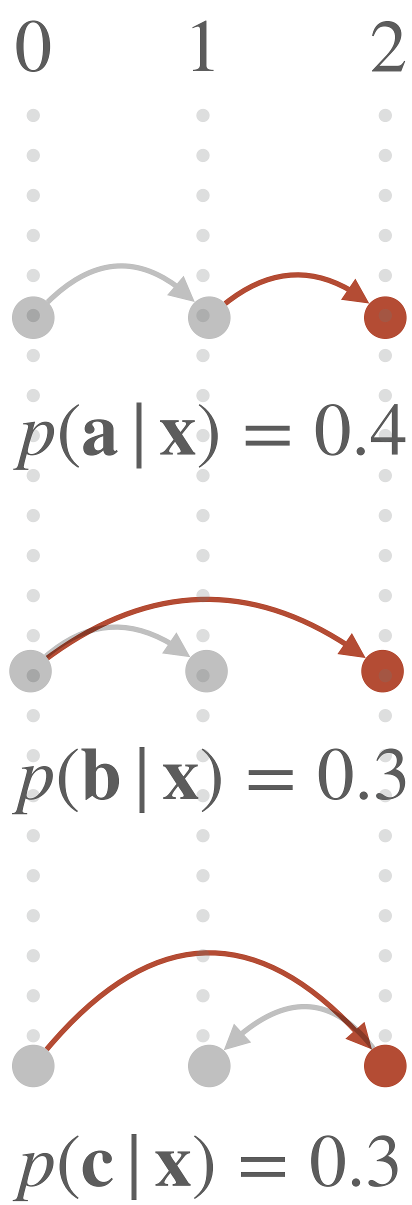


$\hat{\mathbf{w}}$	0	1	2
0	/	$\log 0.7$	?
1	/	/	?
2	/	$\log 0.3$	/

Focus on vertex 1

- Probability to have vertex 0 as head: $p(\mathbf{a} \mid \mathbf{x}) + p(\mathbf{b} \mid \mathbf{x}) = 0.4 + 0.3 = 0.7$
- Probability to have vertex 2 as head: $p(\mathbf{c} \mid \mathbf{x}) = 0.3$

TOKEN-SEPARABLE LOSS FUNCTIONS

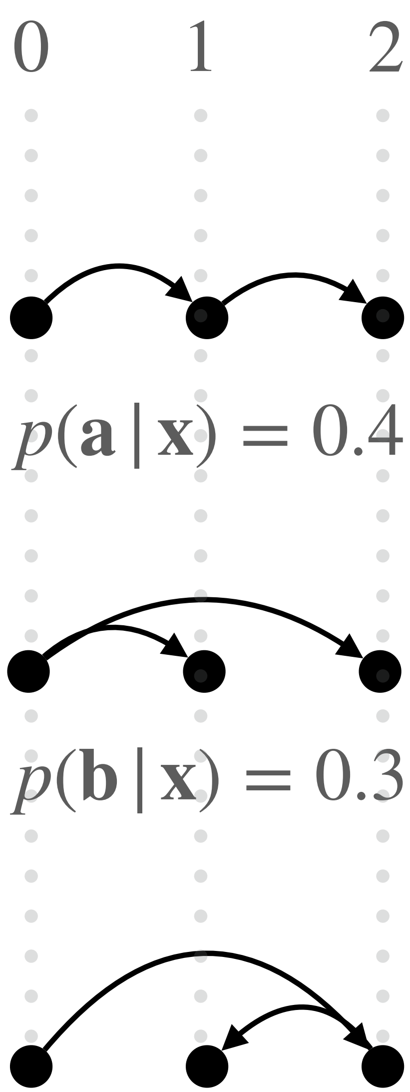


$\widehat{\mathbf{w}}$	0	1	2
0	/	$\log 0.7$	$\log 0.6$
1	/	/	$\log 0.4$
2	/	$\log 0.3$	/

Focus on vertex 2

- Probability to have vertex 0 as head: $p(\mathbf{b} \mid \mathbf{x}) + p(\mathbf{c} \mid \mathbf{x}) = 0.3 + 0.3 = 0.6$
- Probability to have vertex 1 as head: $p(\mathbf{a} \mid \mathbf{x}) = 0.4$

TOKEN-SEPARABLE LOSS FUNCTIONS



$p(\mathbf{a}\,|\,\mathbf{x}) = 0.4$

$p(\mathbf{b}\,|\,\mathbf{x}) = 0.3$

$p(\mathbf{c}\,|\,\mathbf{x}) = 0.3$

$\widehat{\mathbf{w}}$	0	1	2
0	/	$\log 0.7$	$\log 0.6$
1	/	/	$\log 0.4$
2	/	$\log 0.3$	/

NOT Bayes consistent :(

$$\widehat{\mathbf{w}}(\mathbf{a}) = \widehat{w}_{0\to1} + \widehat{w}_{1\to2} = \log 0.7 + \log 0.4$$

$$< \log 0.7 + \log 0.6 = \widehat{w}_{0\to1} + \widehat{w}_{0\to2} = \widehat{\mathbf{w}}(\mathbf{b})$$

INTERMEDIATE CONCLUSION

Take home message

Token-separable losses are not necessarily Bayes consistent.

Other examples of separable losses

- Token level NLL for BIO tagging (ignores the fact that a I tag can not follow a O tag)
- Semantic parsing **[Panupong et al., 2019]**
- Discontinuous constituency parsing **[Corro, 2020]**

Should we care about loss function properties?

Machine learning is at the core of modern NLP models, so yes.

Should we care about Bayes consistency?

Clearly, separable losses work in practice, but:

- We need theory, "it works" is not good enough
- Previous work showed that Bayes consistency may be misleading as it ignore the structure of the scoring function **[Long and Servedio, 2013]**

CONCLUSION

CONCLUSION

Take home message 1

Structured prediction is not dead:

- seq-2-seq models are know to fail in several generalization settings (compositional, structural, ...)
- Beside syntactic parsing and alignment models for MT,
  there are many NLP problems for which combinatorial algorithms have been understudied.
  See for example [Corro, 2022] for NER
- Open question: how to embed "structural knowledge" in seq-2-seq models?

obvious exaggeration :)

Take home message 2

- Loss functions are the cornestone of machine learning
- NLP has a lot of interesting learning problems were theory is missing

For other examples in NLP, check: [Ma & Collins, 2018] [Effland & Collins, 2021]

(advertising) Book on discrete latent structure in neural networks

https://arxiv.org/abs/2301.07473

Discrete Latent Structure in Neural Networks

Vlad Niculae[1], Caio F. Corro[2], Nikita Nangia[3], Tsvetomila Mihaylova[4,5] and André F. T. Martins[4,5,6]