New Algorithms for Semantics-Based Machine Translation

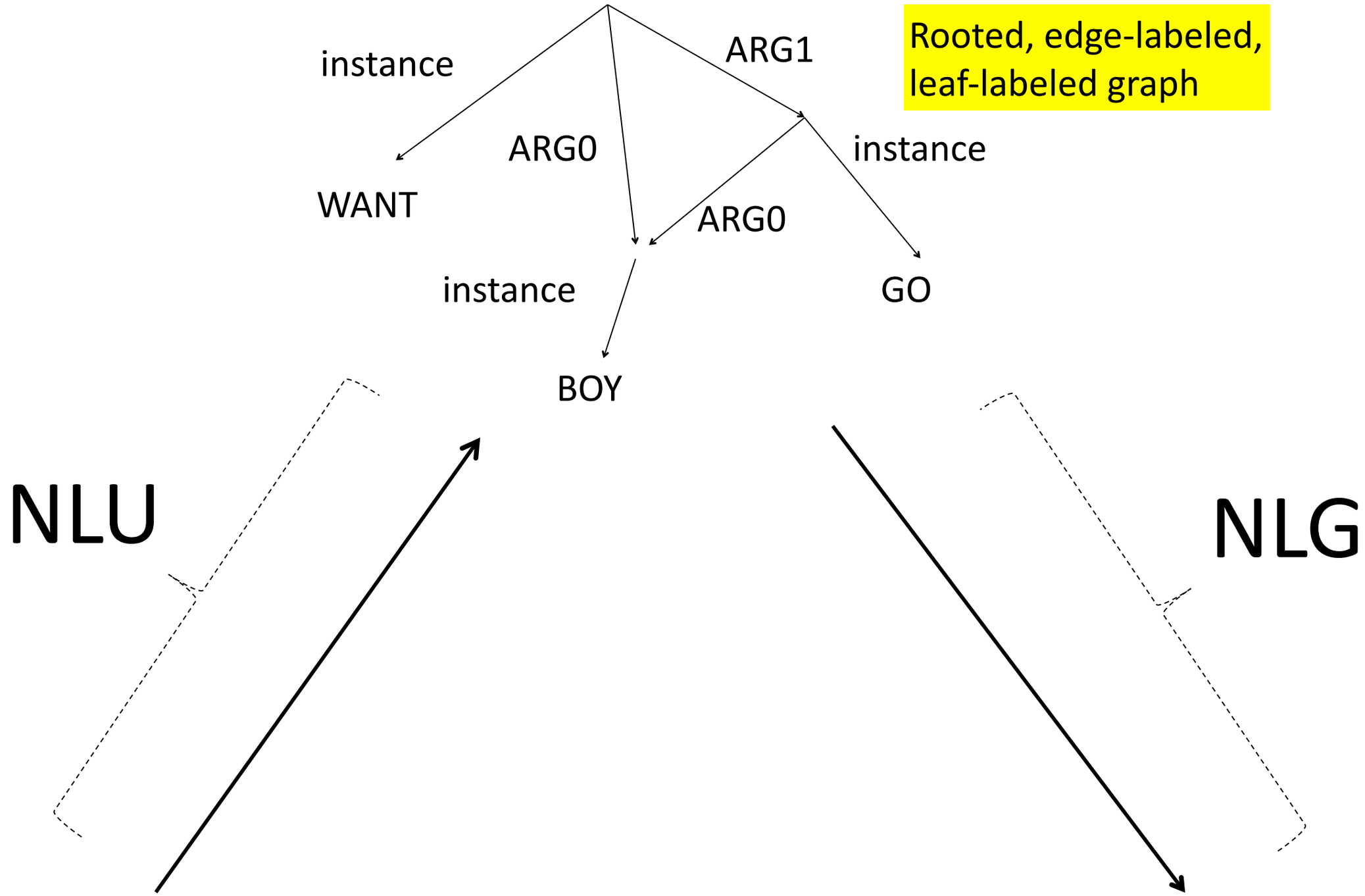
USC/ISI

2012 November 2

USC University of Southern California

Why Semantics?

- Explicit meaning representation will improve meaning-preserving translation
- Modeling meaning instead of surface realizations
  → learn more from less data


instance
ARG1
Rooted, edge-labeled,
leaf-labeled graph
ARG0
instance
WANT
ARG0
instance
GO
BOY
NLU
NLG
source
string
target
string
The boy wants to go.
男孩子想去。

Five Equivalent Meaning Representation Formats

"The boy wants to go."

LOGICAL FORM

∃w, b, g : instance(w, WANT) ^
instance(g, GO) ^
instance(b, BOY) ^
agent(w, b) ^
patient(w, g) ^
agent(g, b)

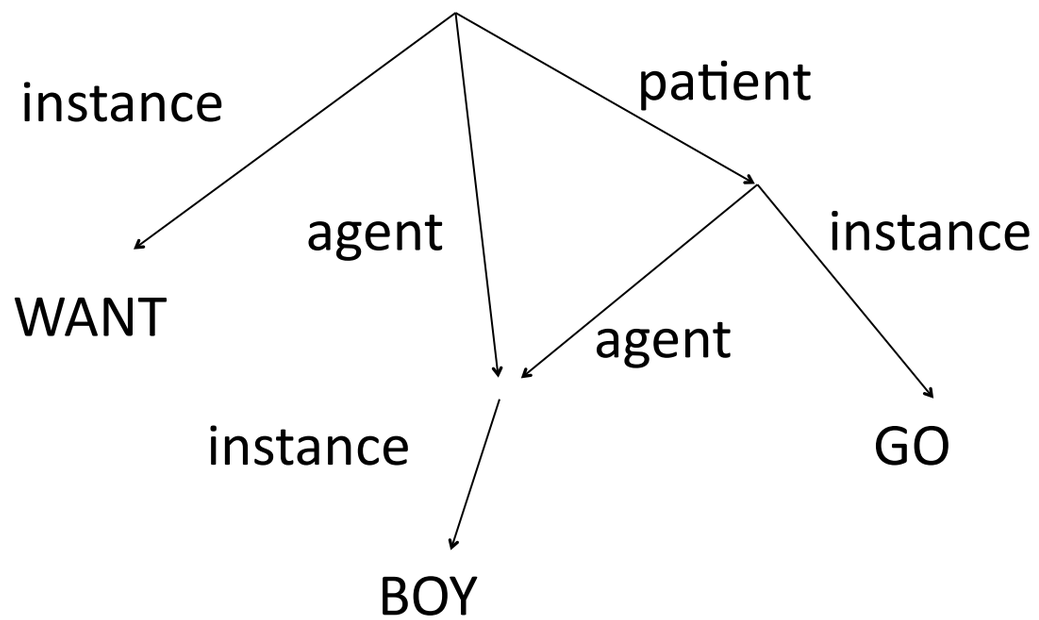
PENMAN

```
(w / WANT
  :agent (b / BOY)
  :patient (g / GO
             :agent b)))
```

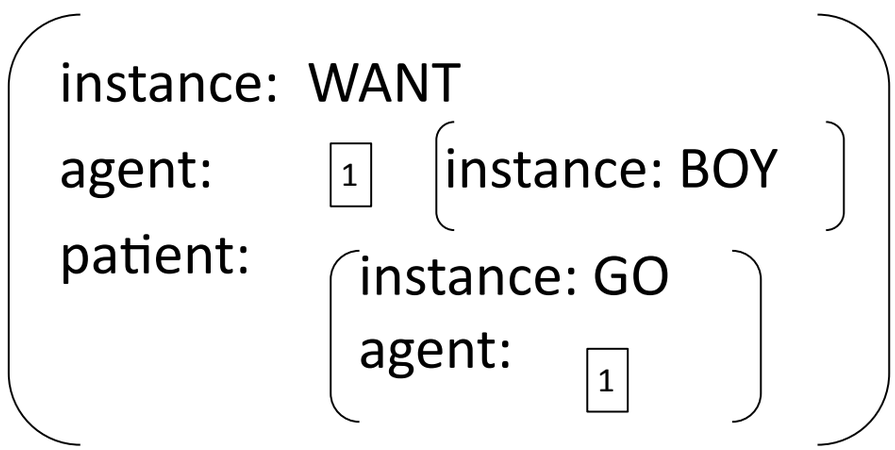
PATH EQUATIONS

((x0 instance) = WANT
((x1 instance) = BOY
((x2 instance) = GO
((x0 agent) = x1
((x0 patent) = x2
((x2 agent) = x1

DIRECTED ACYCLIC GRAPH



FEATURE STRUCTURE

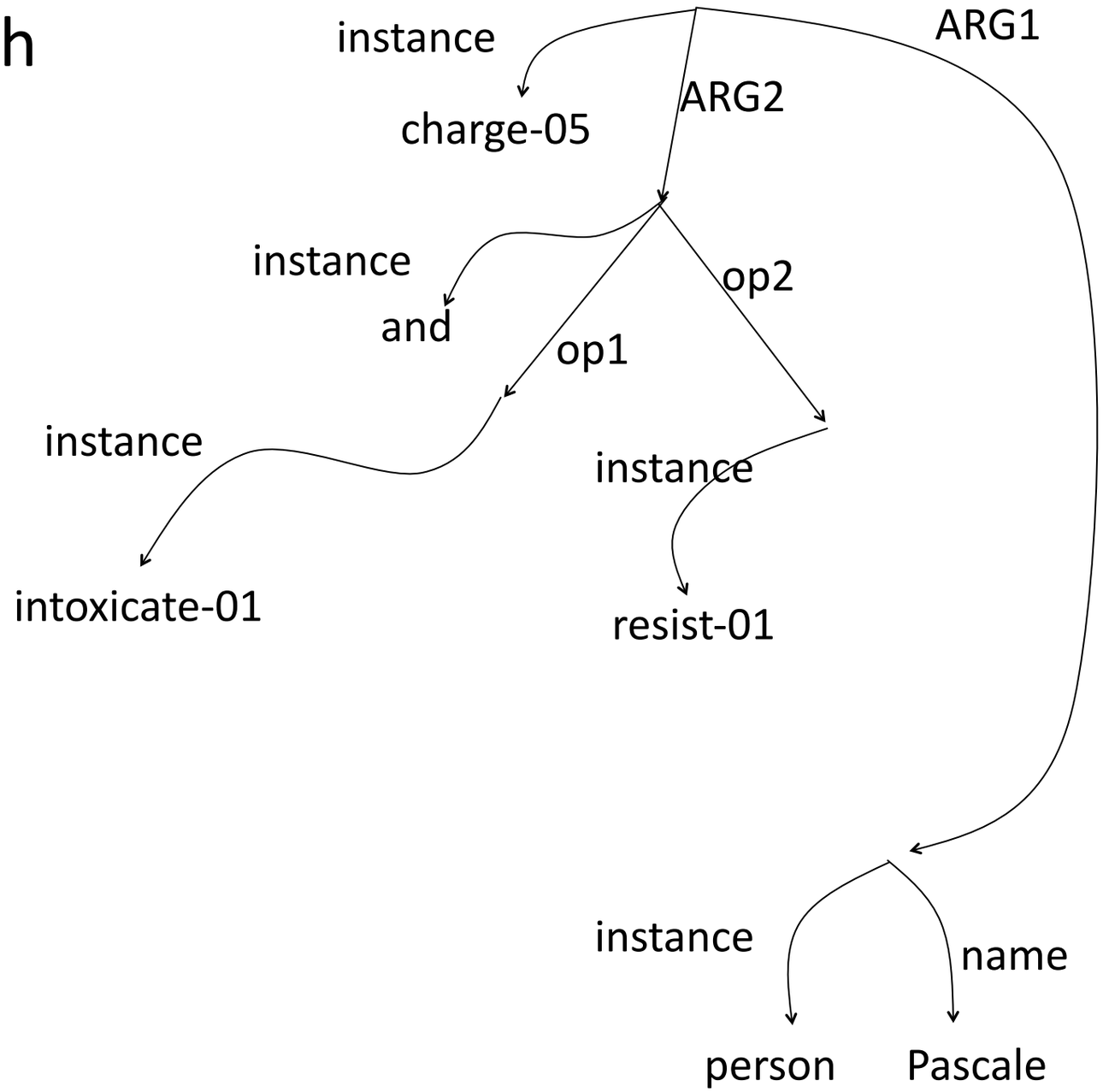


Multiple Roles → Graph Structure

"Pascale was charged with public intoxication and resisting arrest."

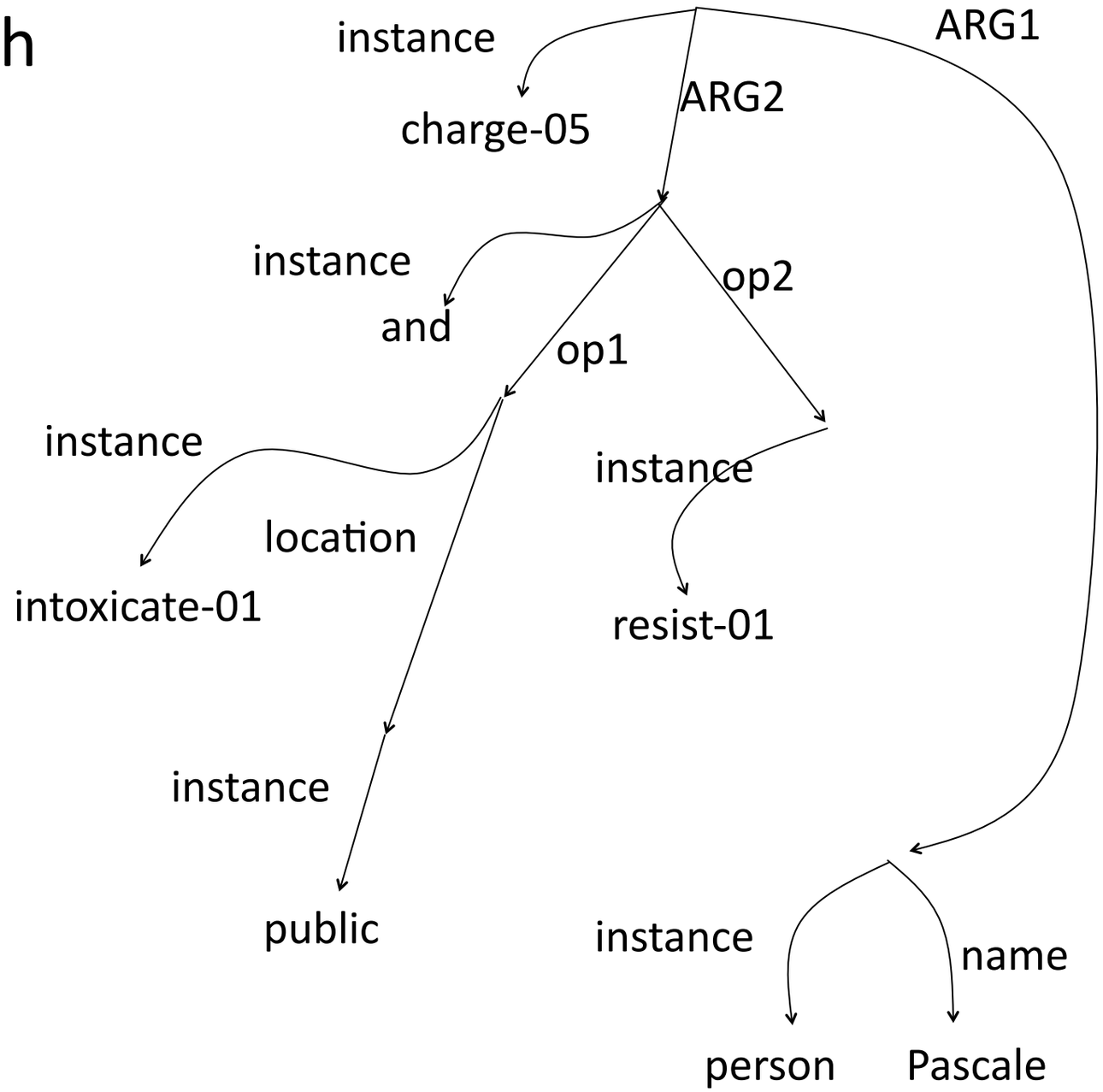
Multiple Roles → Graph Structure

"Pascale was charged with public intoxication and resisting arrest."



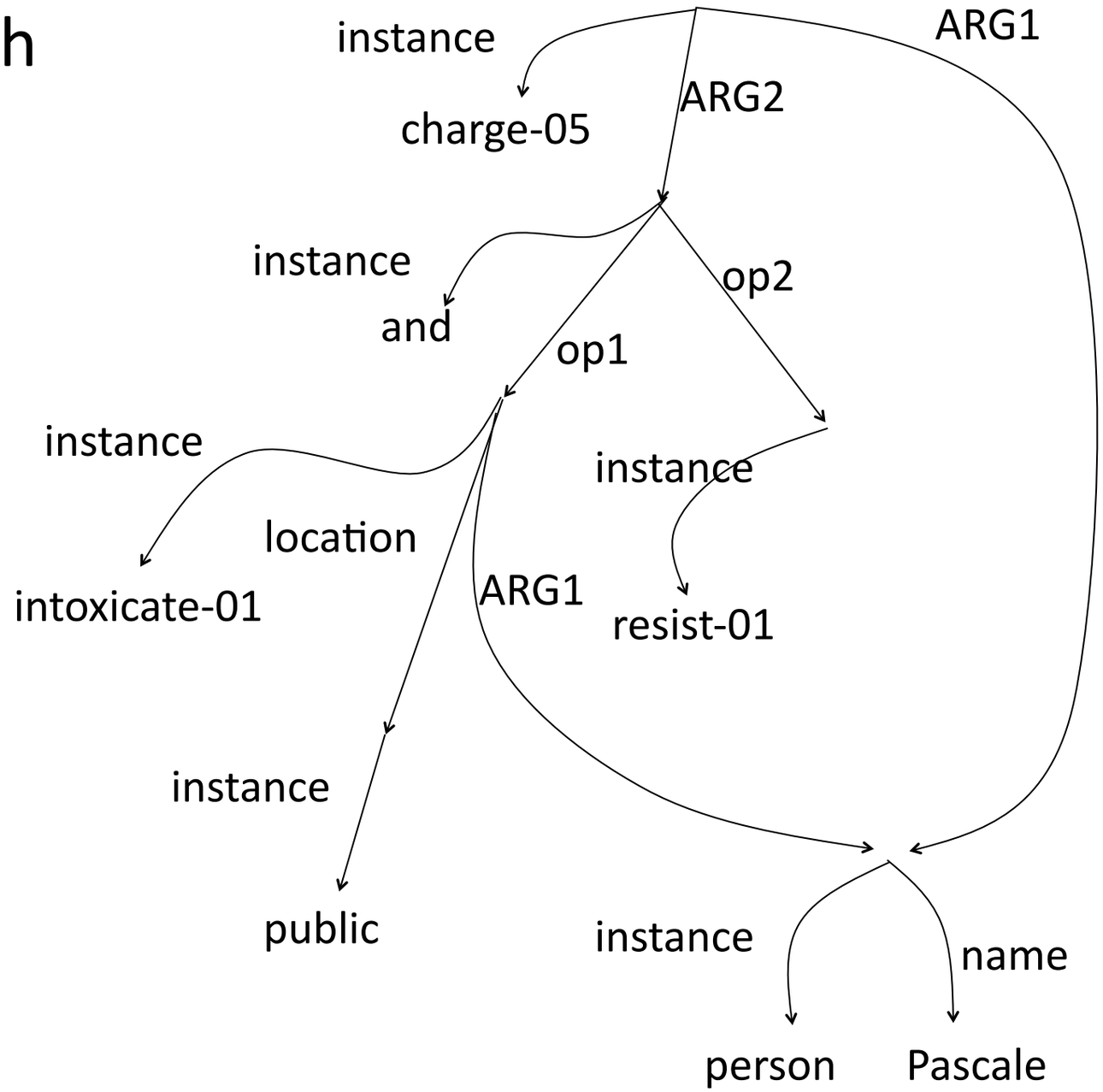
Multiple Roles → Graph Structure

"Pascale was charged with public intoxication and resisting arrest."



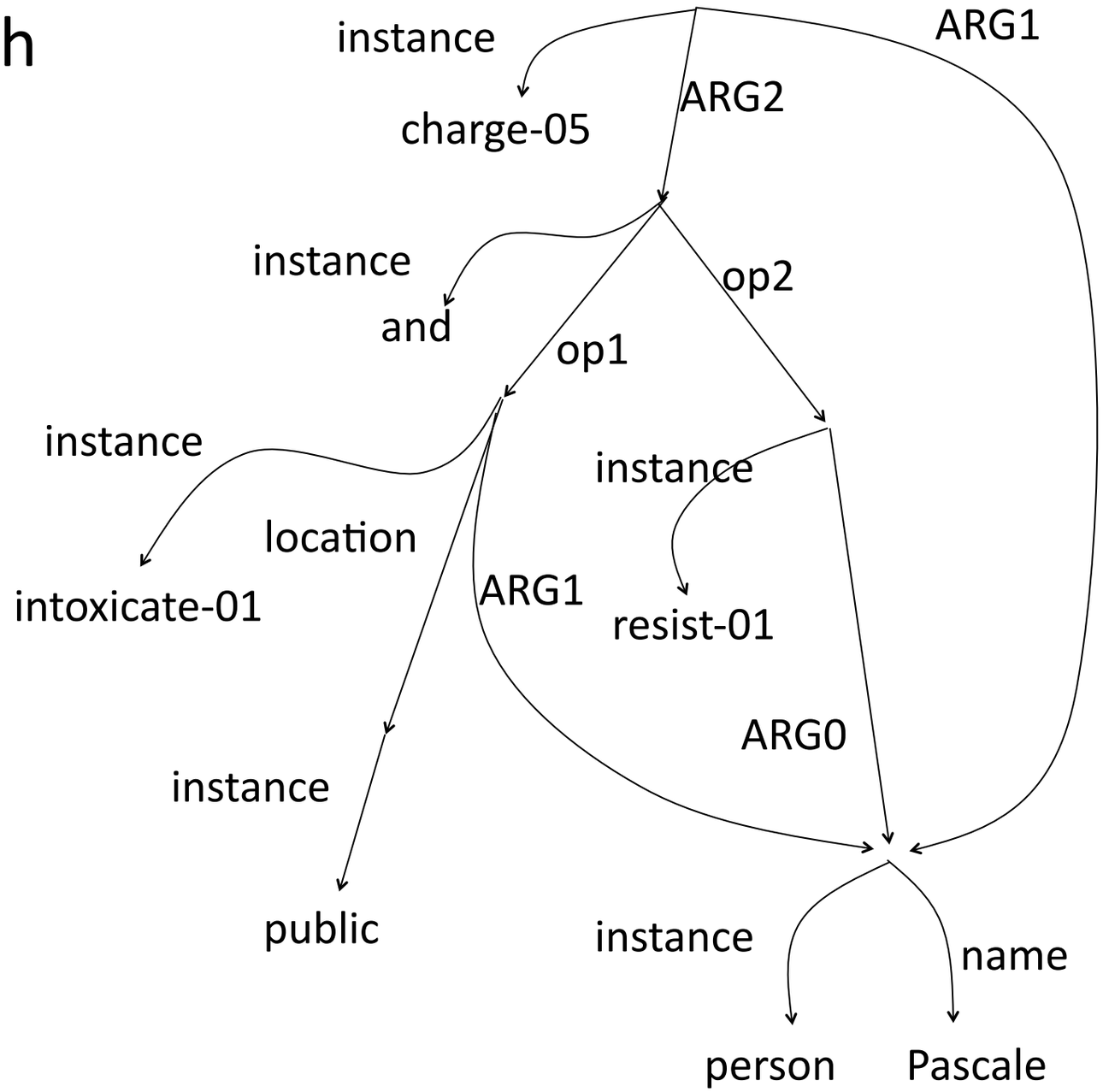
Multiple Roles → Graph Structure

"Pascale was charged with public intoxication and resisting arrest."



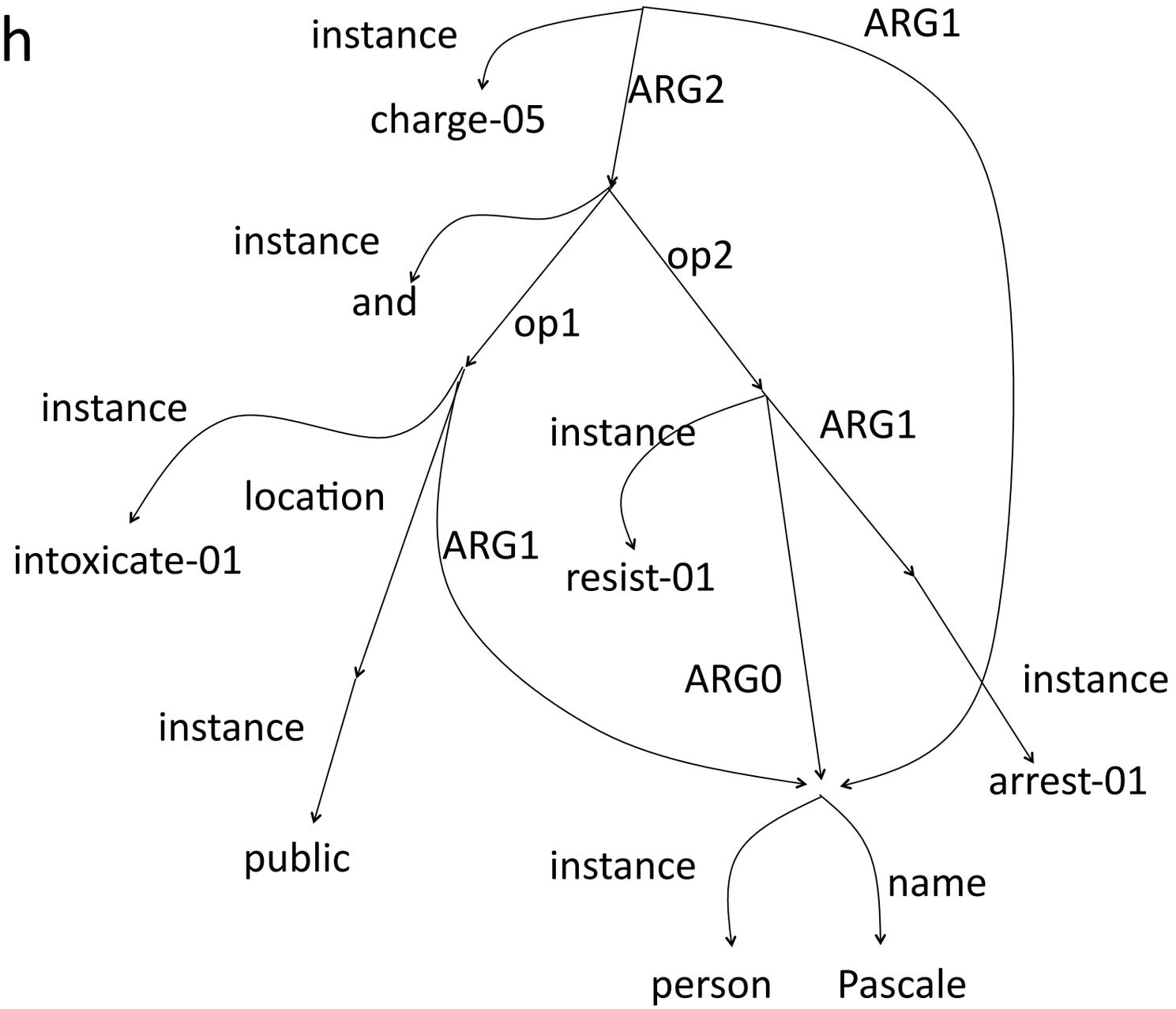
Multiple Roles → Graph Structure

"Pascale was charged with public intoxication and resisting arrest."



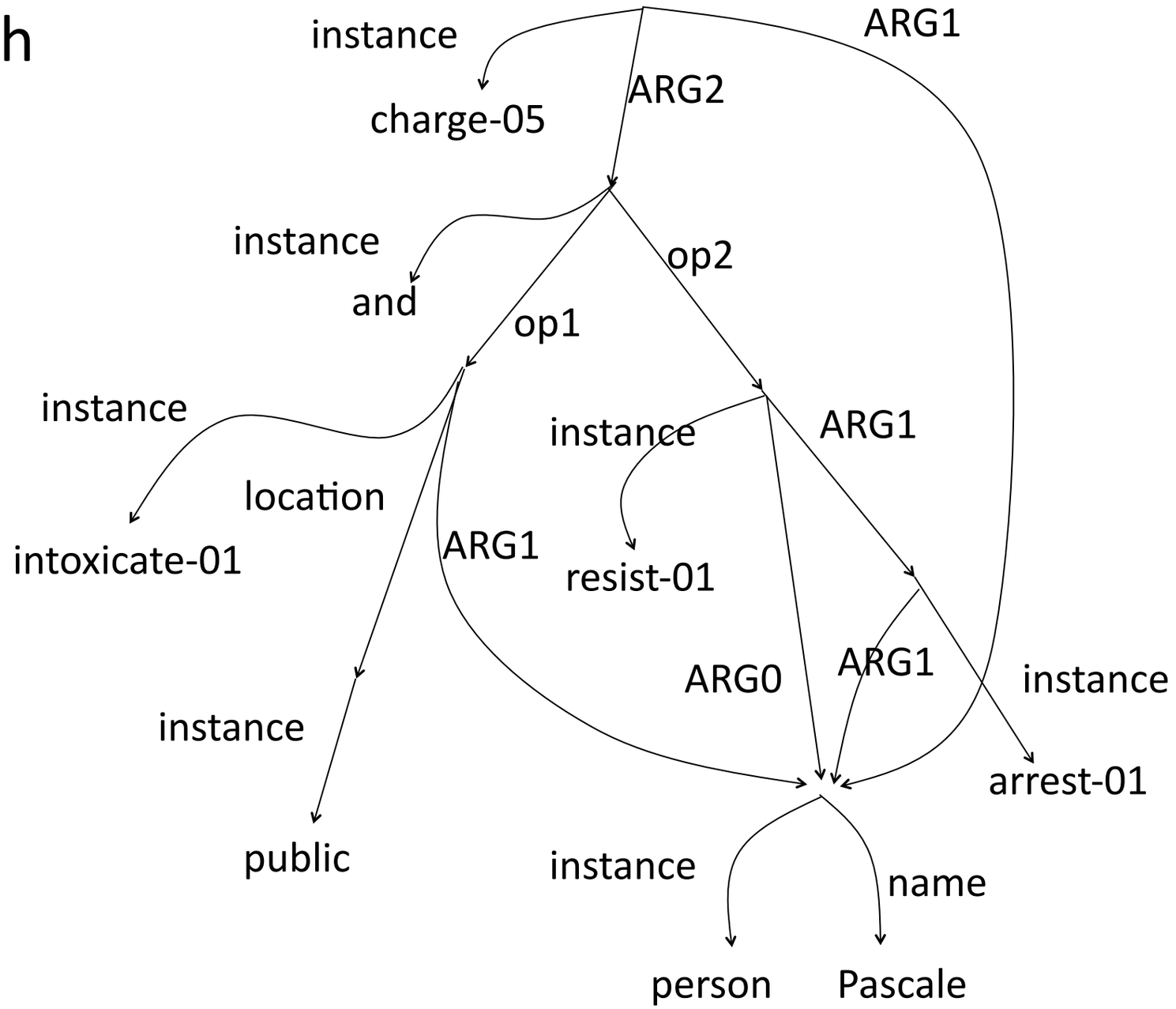
Multiple Roles → Graph Structure

"Pascale was charged with public intoxication and resisting arrest."



Multiple Roles → Graph Structure

"Pascale was charged with public intoxication and resisting arrest."



Meaning-based MT

- Too big for just this MURI:
  - –What content goes into the meaning representation?
    - linguistics, annotation (Nathan Schneider, CMU)
  - –How are meaning representations probabilistically generated, transformed, scored, ranked?
    - automata theory, efficient algorithms
  - –How can a full MT system be built and tested?
    - engineering, language modeling, features, training

AMR Editor - knight - Mozilla Firefox

File Edit View History Bookmarks Tools Help

AMR Editor - knight

www.isi.edu/cgi-bin/div3/mt/load-amr-v1.6.cgi

AMR Editor knight Older versions: 1.2 1.3 1.4 1.5

Written by Ulf Hermjakob, USC/ISI Version 1.6 June 19, 2012

Sentence: Pierre Vinken , 61 years old , will join the board as a nonexecutive director Nov. 29 . ON

```
(j / join-01
  :ARG0 (p / person
           :name (p2 / name
                    :op1 "Pierre"
                    :op2 "Vinken")
           :age (t / temporal-quantity
                   :unit (y / year)
                   :quant 61))
  :ARG1 (b / board)
  :prep-as (d2 / director
              :mod (e / executive
                      :polarity -))
  :time (d / date-entity
           :month 11
           :day 29))
```

Abstract Meaning Representation

- 35-page guidelines.
- Extensive use of PropBank predicates, but cover all words in sentence.
- AMR Editor with logins and worksets.
- 7 minutes per sentence, enabling very large-scale Sembanking.

Enter text command: QuickRef

Last command: Load AMR from workset list

Or select an action template: top add add-ne replace delete move undo exit/load check props exit/save options clear help gene

Workset wsj100-sent 1/100 ◄ nw.wsj_0001.1 (saved) ► Next: nw.wsj_0001.2 sent. menu export

---

Log: initialized empty AMR
For role checking, loaded 99 roles and 6 non-roles.
For OntoNotes frame availability check, loaded 5528 verbs.

start amadeus:/nfs... cage:/nfs/isd3... cage:/nfs/isd3... tutorials to-do.txt - No... Inbox - Mozilla... tag-plus-12.pptx AMR Editor - k... EN 9:17 AM

Meaning-based MT

- Too big for just this MURI:
  - What content goes into the meaning representation?
    - linguistics, annotation
  - How are meaning representations probabilistically generated, transformed, scored, ranked?
    - automata theory, efficient algorithms
  - How can a full MT system be built and tested?
    - engineering, language modeling, features, training

General-Purpose Algorithms for Feature Structures (Graphs)

	String World (words)	Tree World (syntax)	Graph World? (semantics)
Acceptor	Finite-state acceptors	Tree automata	
Transducer	Finite-state transducers	Tree transducers	
Membership checking	$O(n)$	$O(n)$ for trees $O(n^3)$ for strings	
N-best …	… paths through an WFSA (Viterbi, 1967; Eppstein, 1998)	… trees in a weighted forest (Jiménez & Marzal, 2000; Huang & Chiang, 2005)	
EM training	Forward-backward EM (Baum/ Welch, 1971; Eisner 2002)	Tree transducer EM training (Graehl & Knight, 2004)	
Intersection	WFSA intersection	Tree acceptor intersection	
Transducer composition	WFST composition (Pereira & Riley, 1996)	Many tree transducers not closed under composition (Maletti et al 09)	
General tools	Carmel, OpenFST	Tiburon (May & Knight 10)	

Hyperedge Replacement Grammars

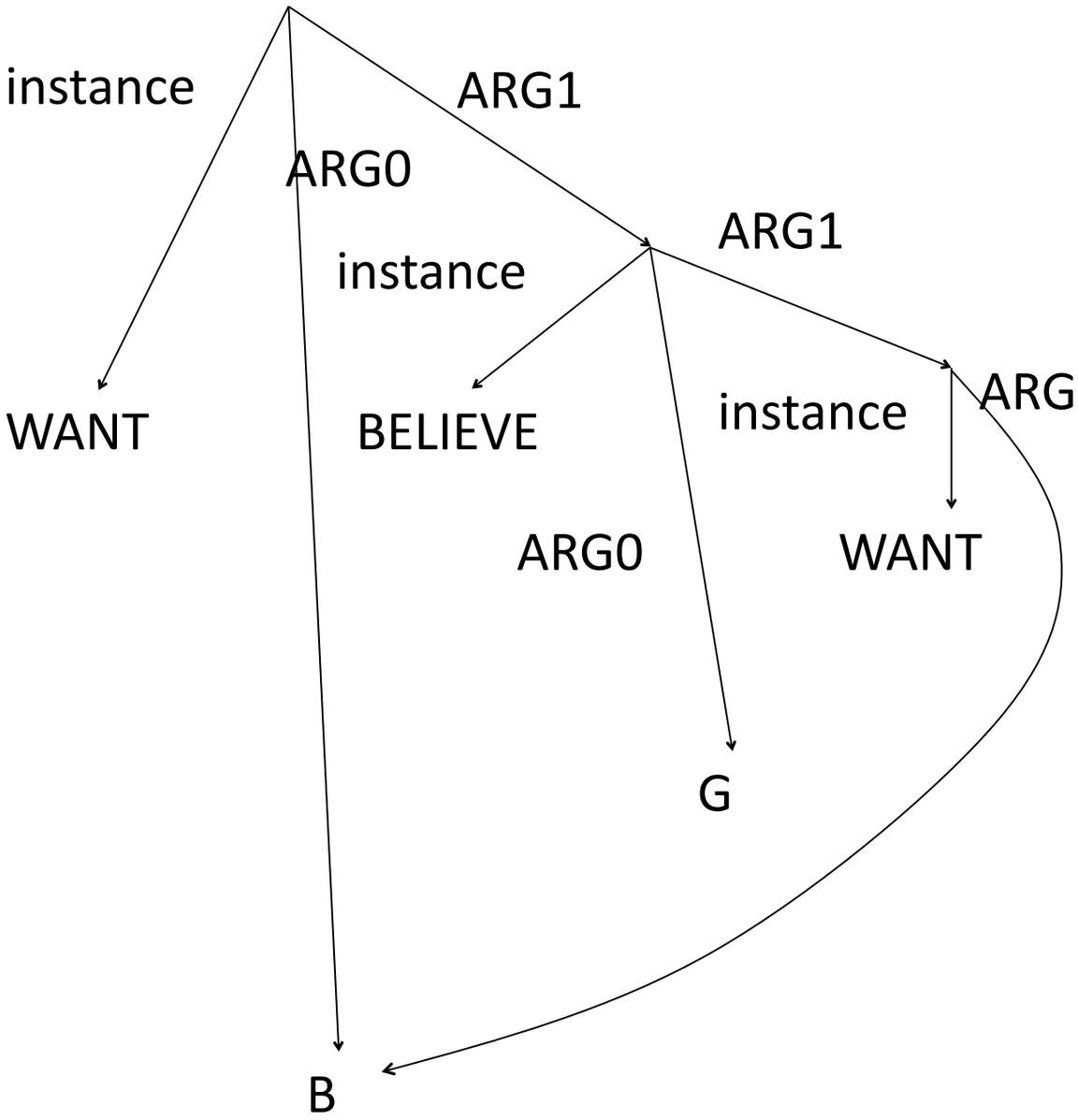
- Survey: Drewes et al., 1997

- Several NLP-related publications from USC/ISI forthcoming

- Key idea: context-free rewriting

HRG Derivation

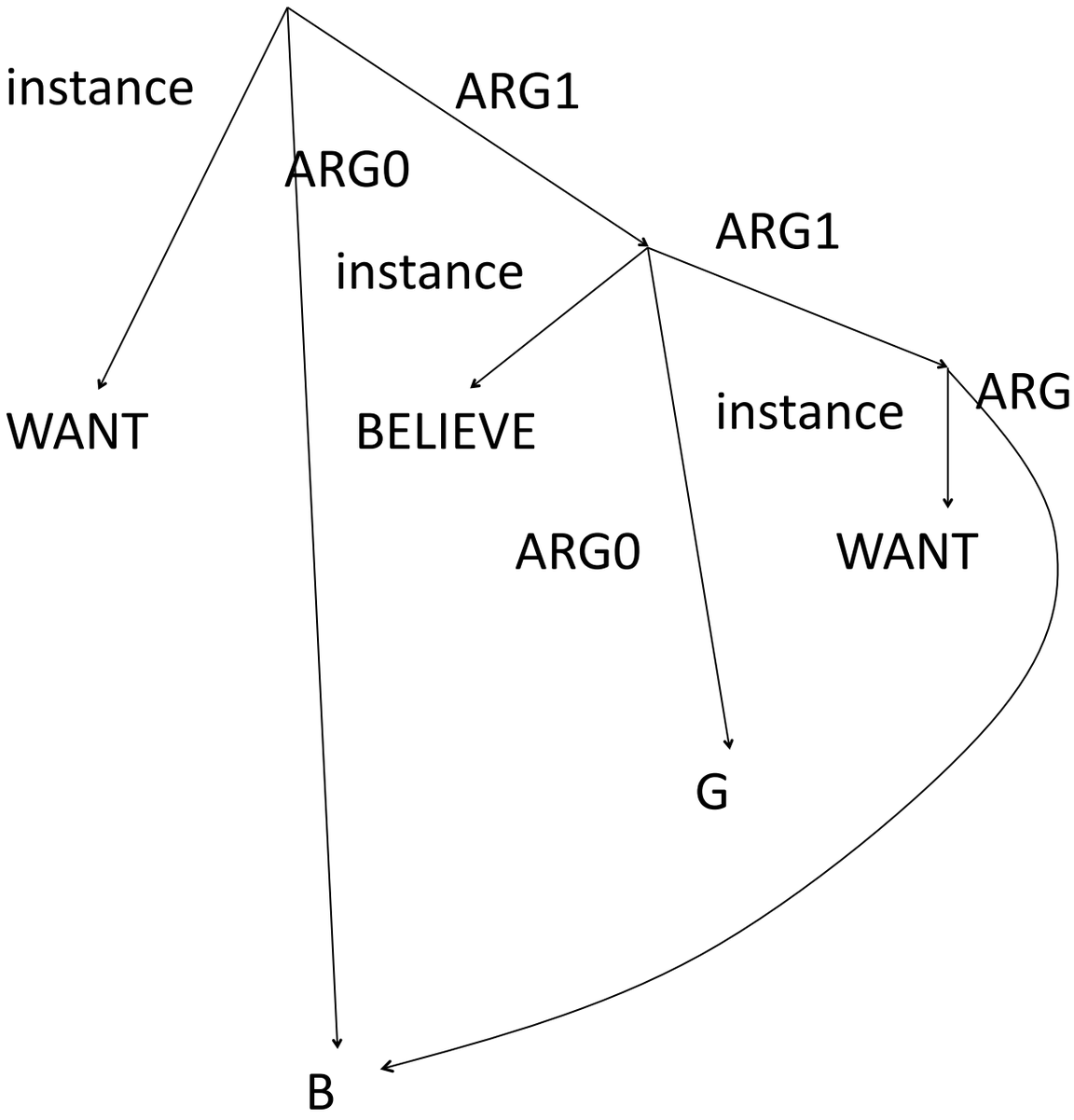
LET'S DERIVE THIS:

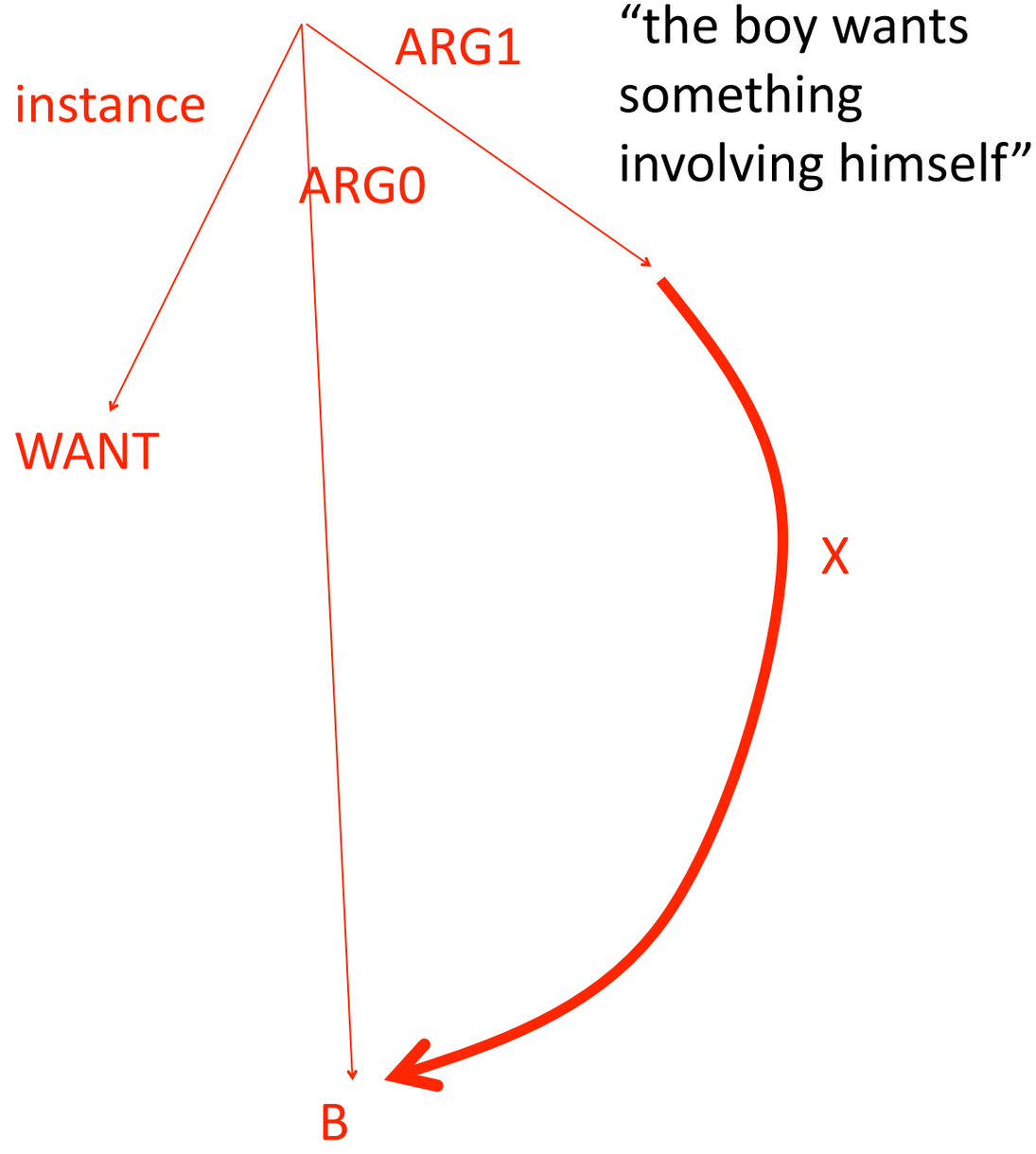


= boy wants girl to believe that he is wanted

HRG Derivation

LET'S DERIVE THIS:

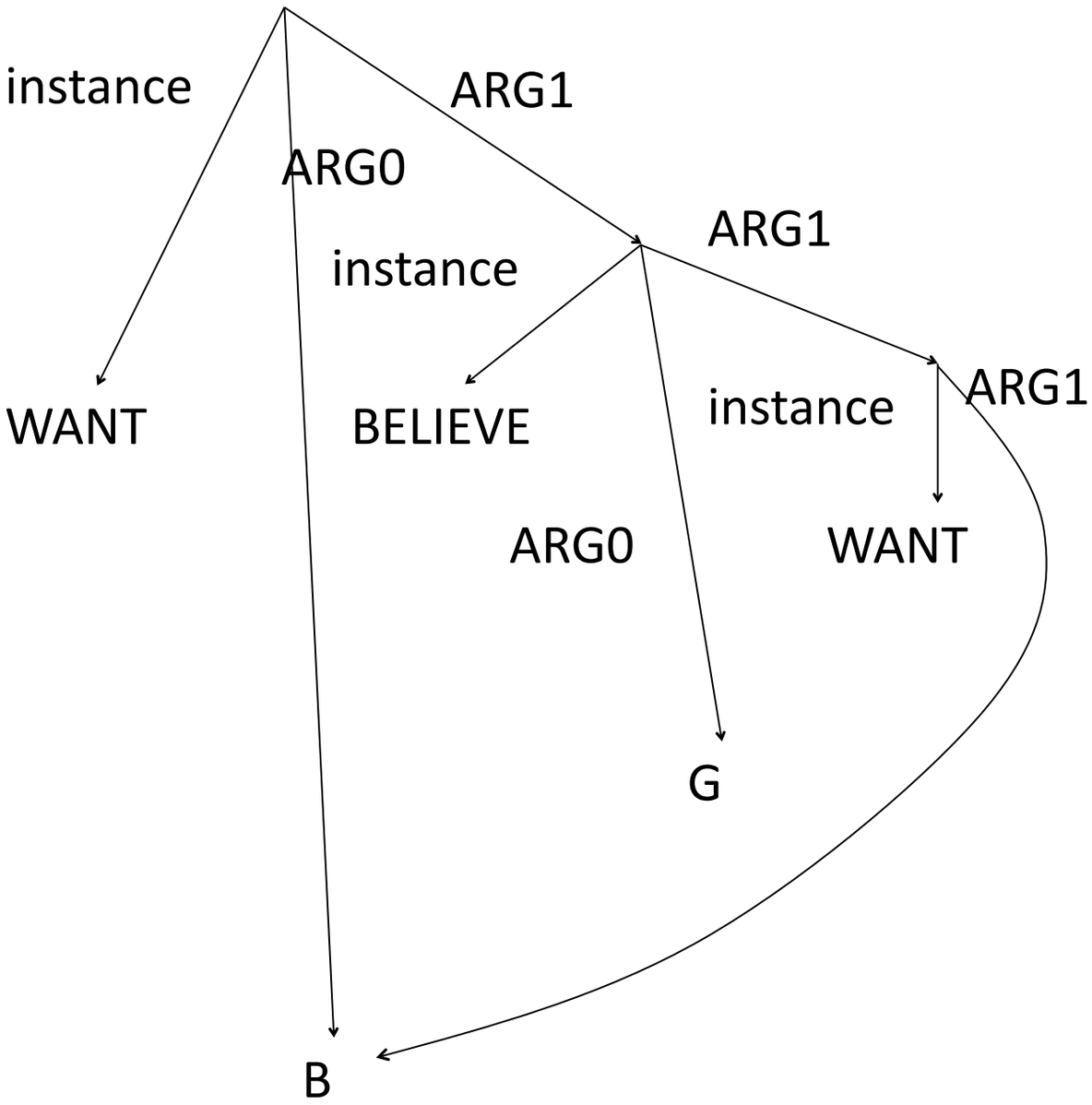


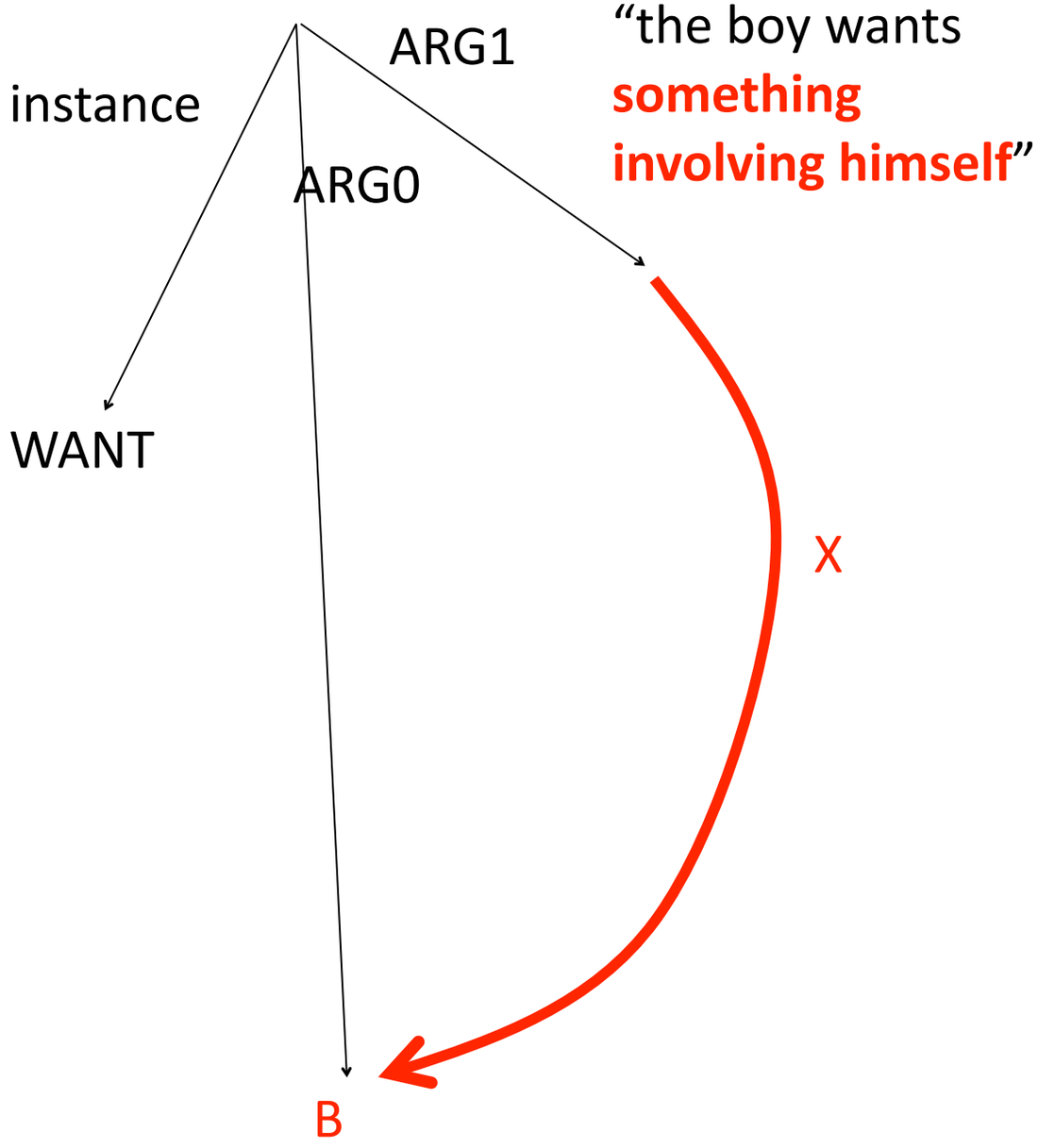


= boy wants girl to believe that he is wanted

HRG Derivation

LET'S DERIVE THIS:



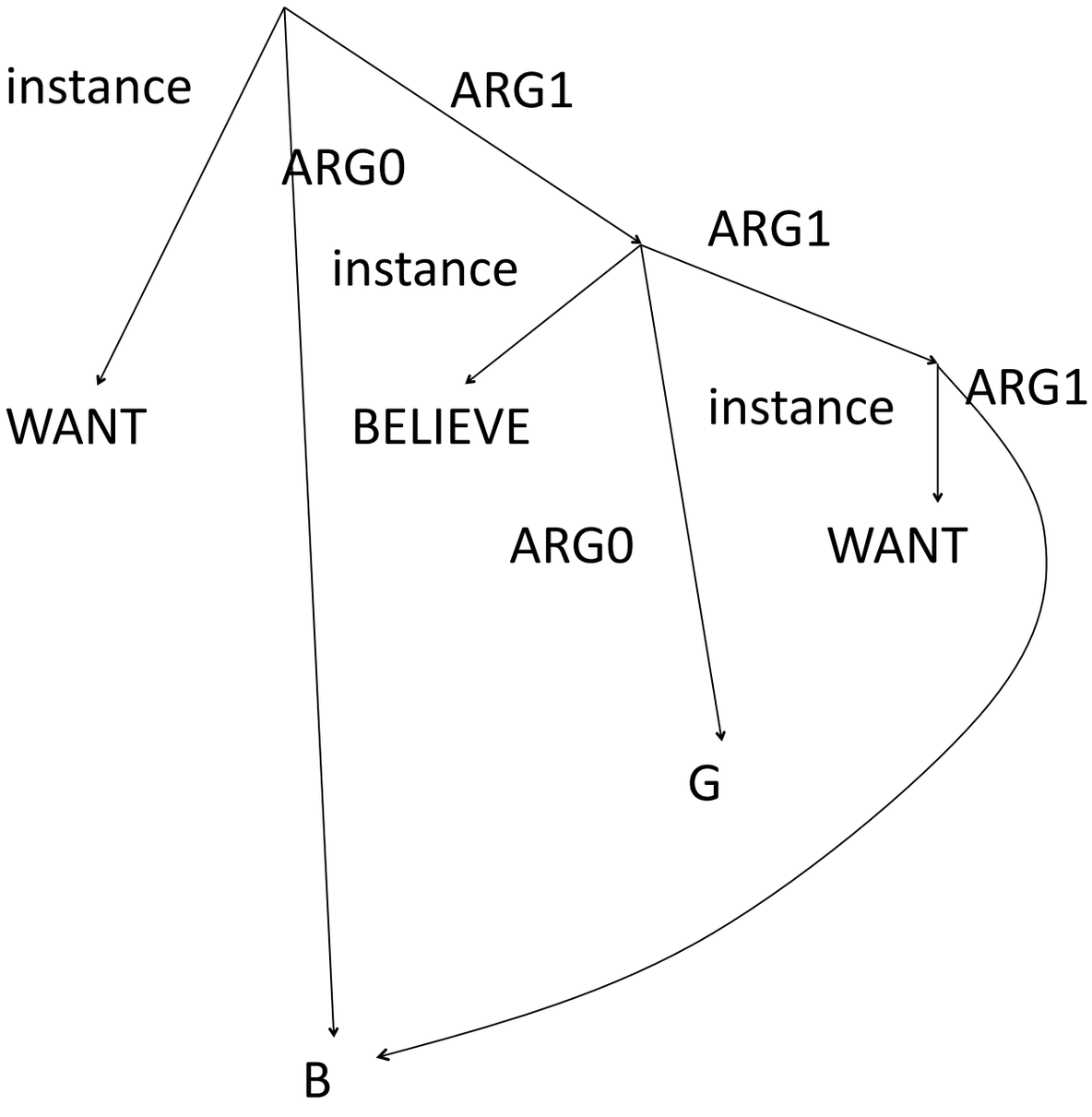


"the boy wants **something involving himself**"

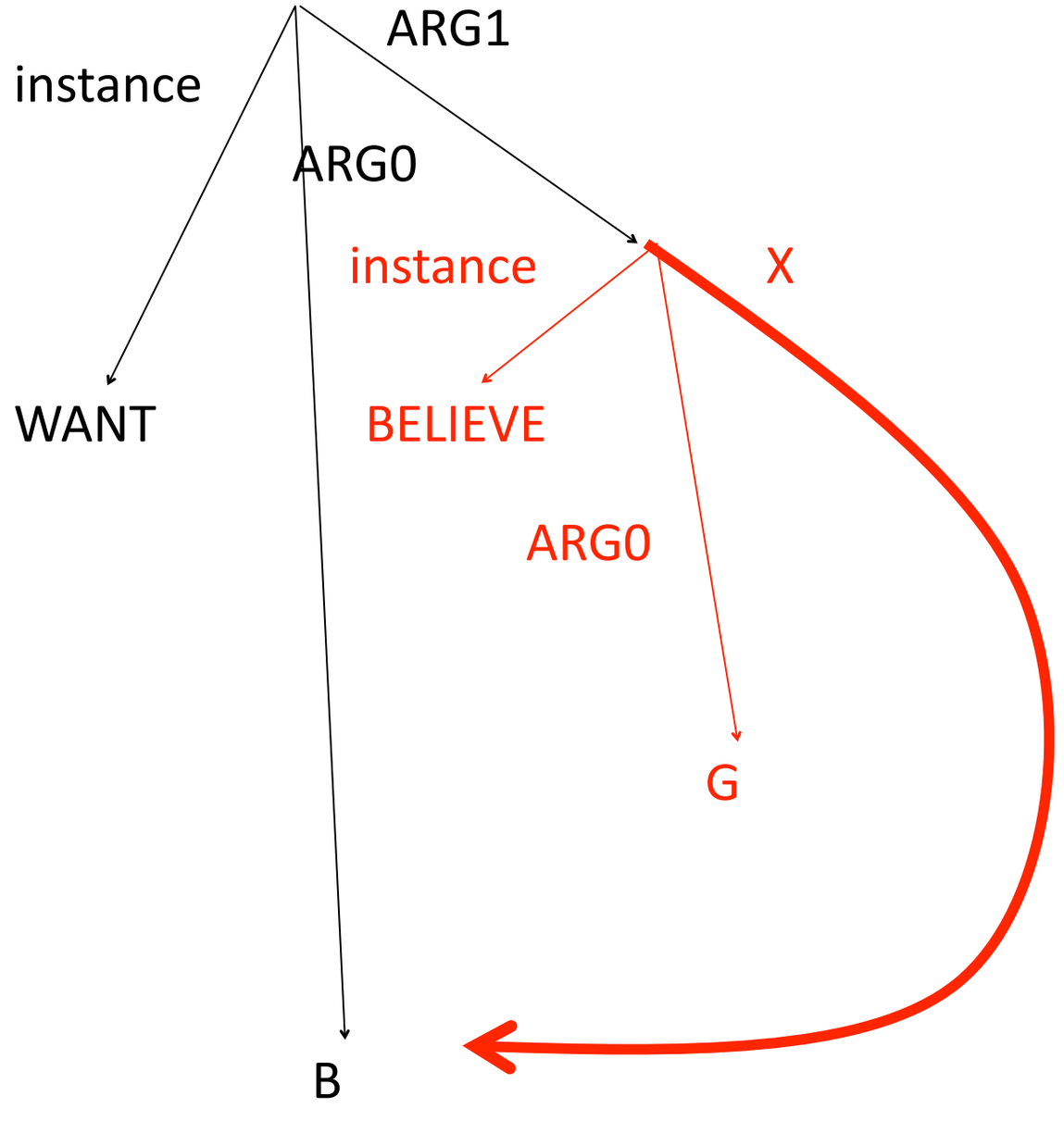
= boy wants girl to believe that he is wanted

HRG Derivation

LET'S DERIVE THIS:



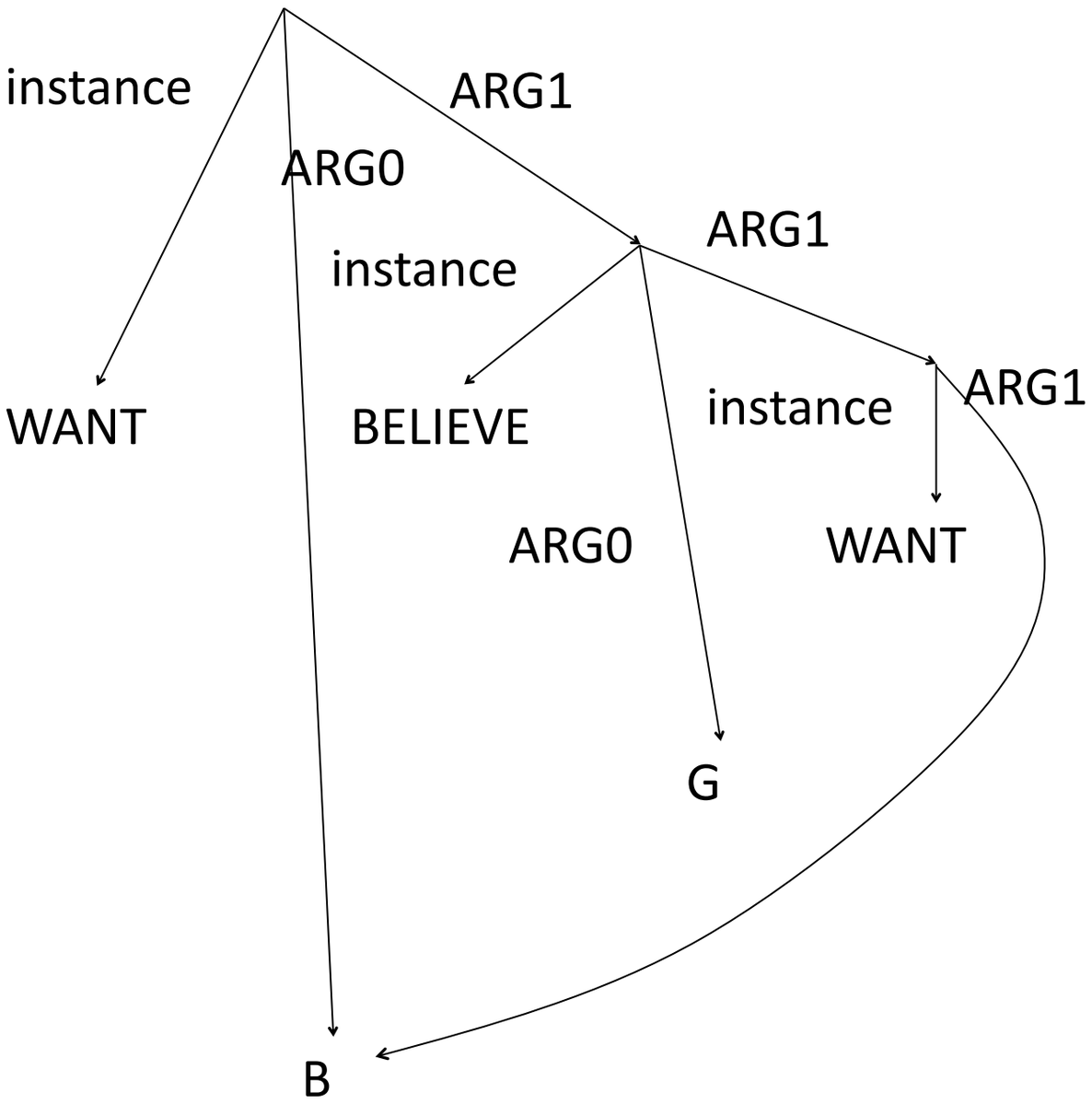
"the boy wants the girl to believe **something involving him**"

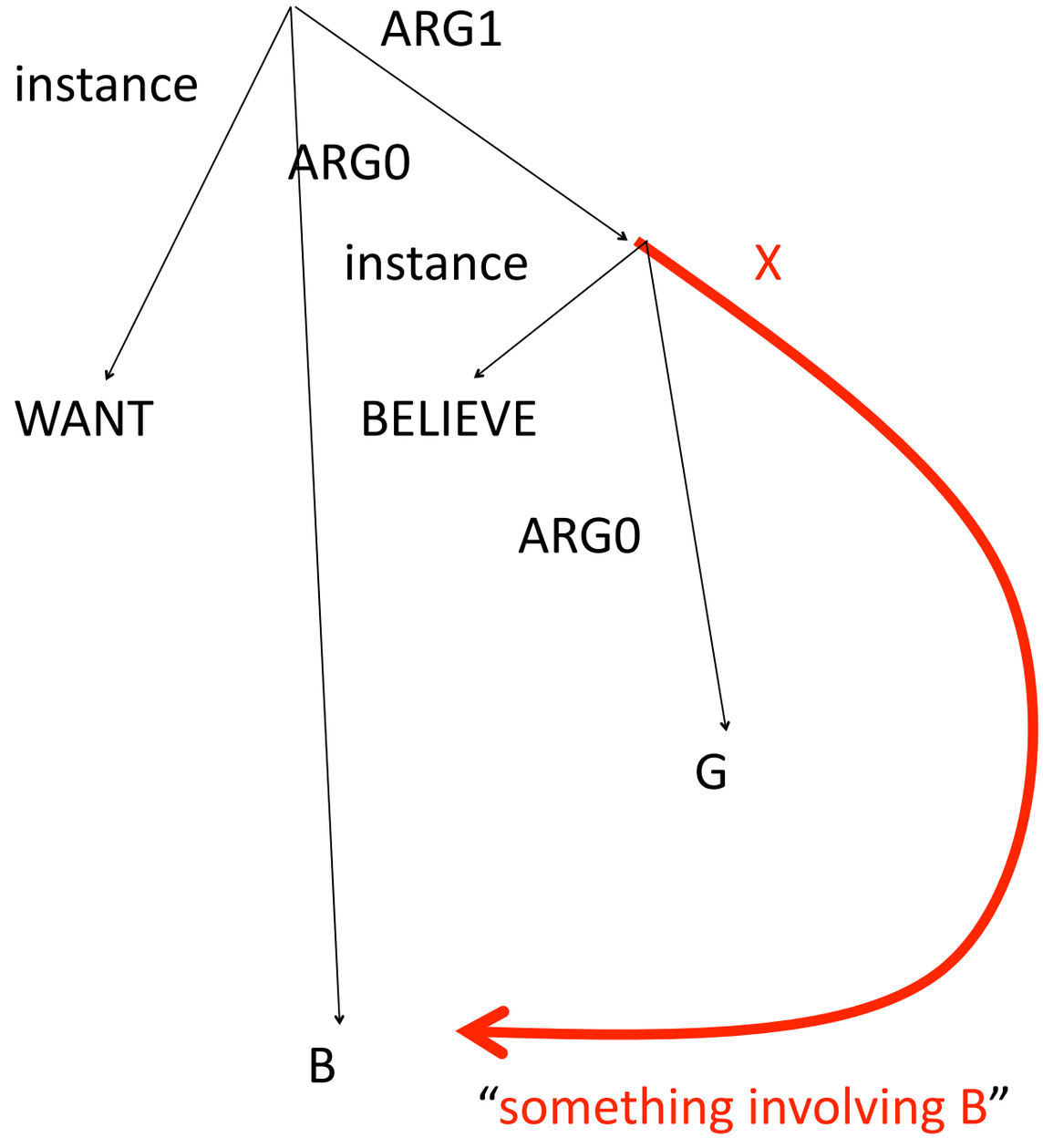


= boy wants girl to believe that he is wanted

HRG Derivation

LET'S DERIVE THIS:

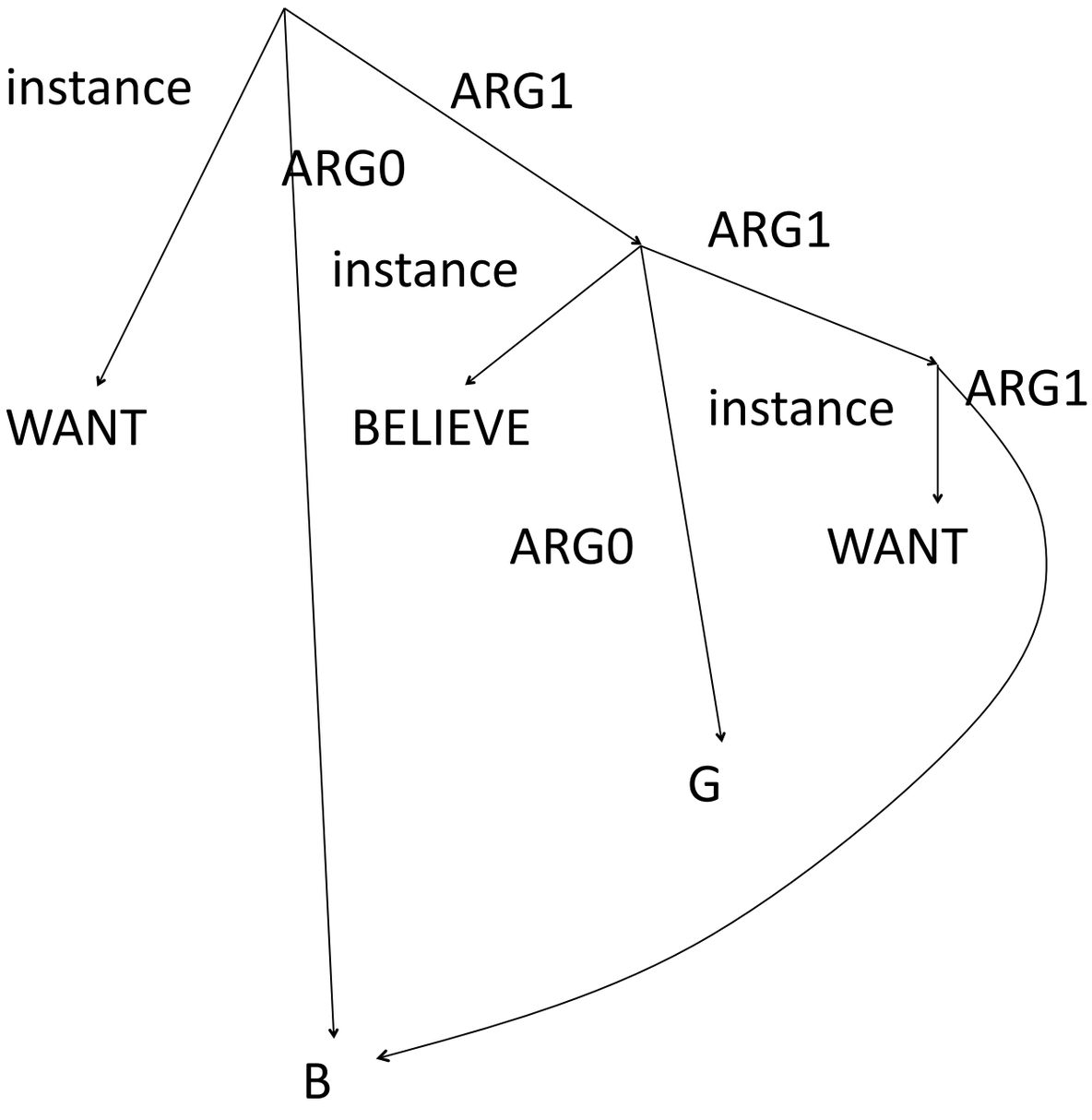


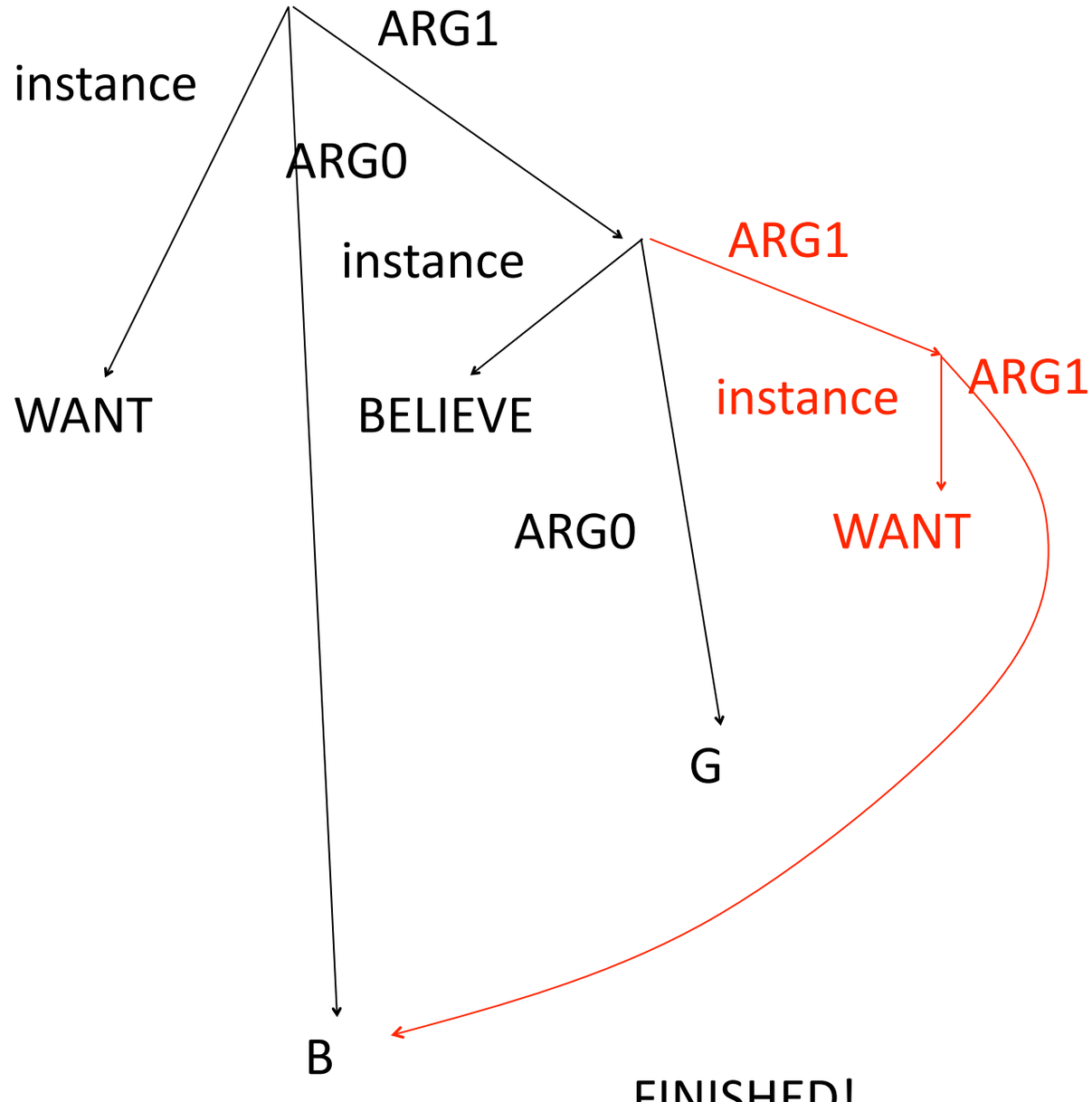


= boy wants girl to believe that he is wanted

HRG Derivation

LET'S DERIVE THIS:



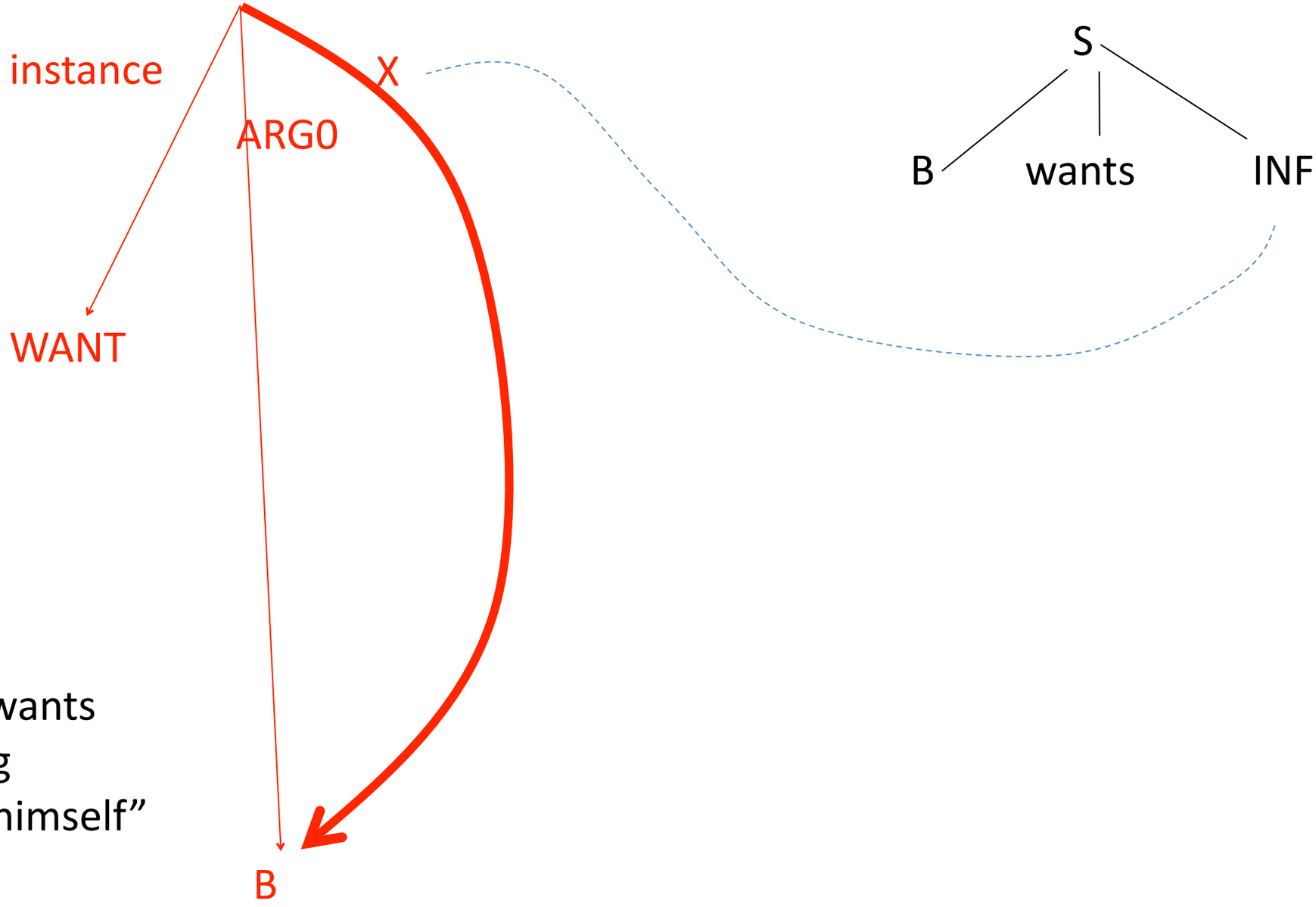


= boy wants girl to believe that he is wanted

General-Purpose Algorithms for Feature Structures (Graphs)

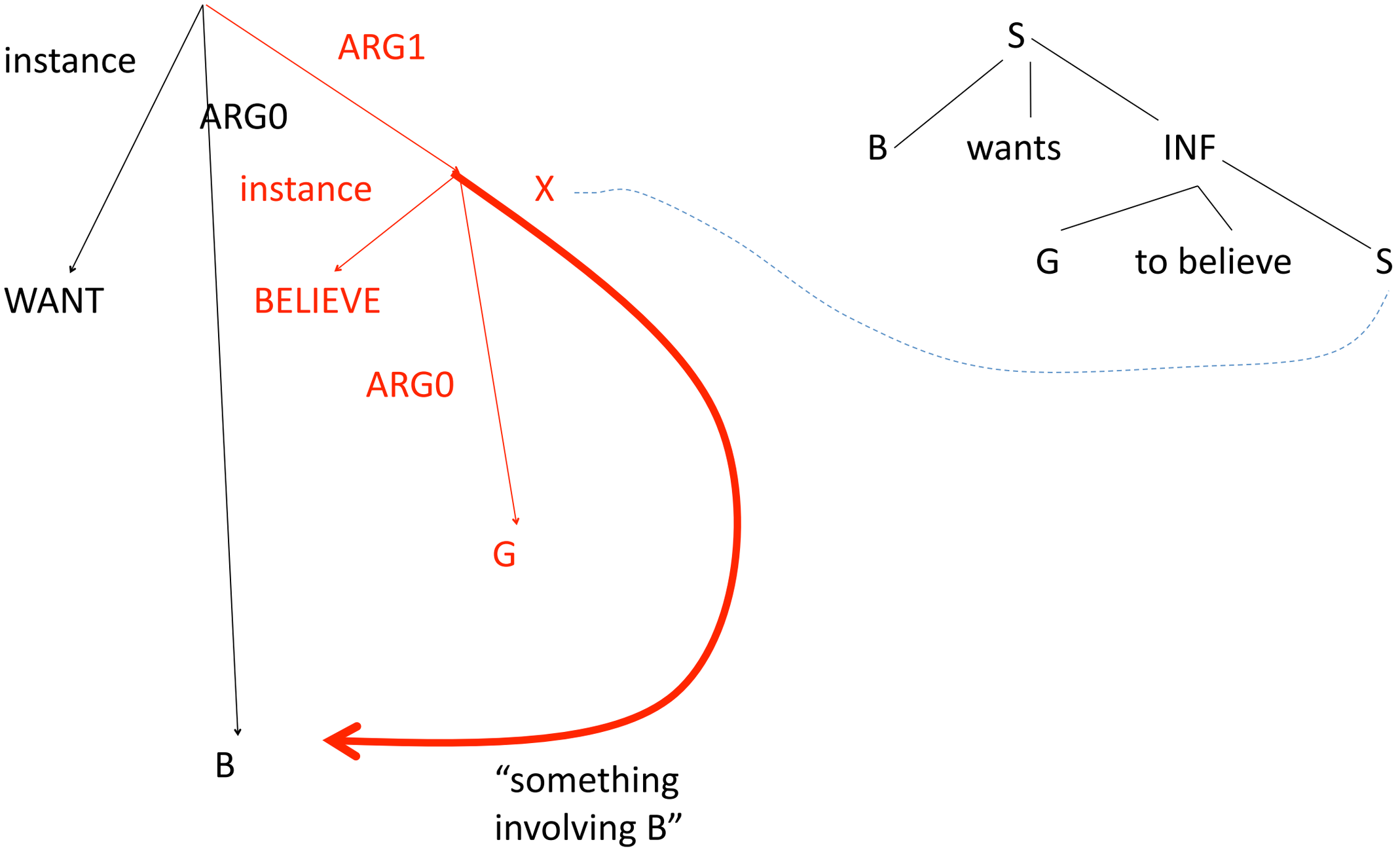
	String World (words)	Tree World (syntax)	Graph World (semantics)
Acceptor	Finite-state acceptors	Tree automata	HRG
Transducer	Finite-state transducers	Tree transducers	Synchronous HRG
Membership checking	$O(n)$	$O(n)$ for trees $O(n^3)$ for strings	$O(n^{k+1})$ for graphs
N-best …	… paths through an WFSA (Viterbi, 1967; Eppstein, 1998)	… trees in a weighted forest (Jiménez & Marzal, 2000; Huang & Chiang, 2005)	… graphs in a weighted forest
EM training	Forward-backward EM (Baum/ Welch, 1971; Eisner 2003)	Tree transducer EM training (Graehl & Knight, 2004)	EM on forests of graphs
Intersection	WFSA intersection	Tree acceptor intersection	Not closed
Transducer composition	WFST composition (Pereira & Riley, 1996)	Many tree transducers not closed under composition (Maletti et al 09)	Not closed
General tools	Carmel, OpenFST	Tiburon (May & Knight 10)	Bolinas

SHRG Derivation

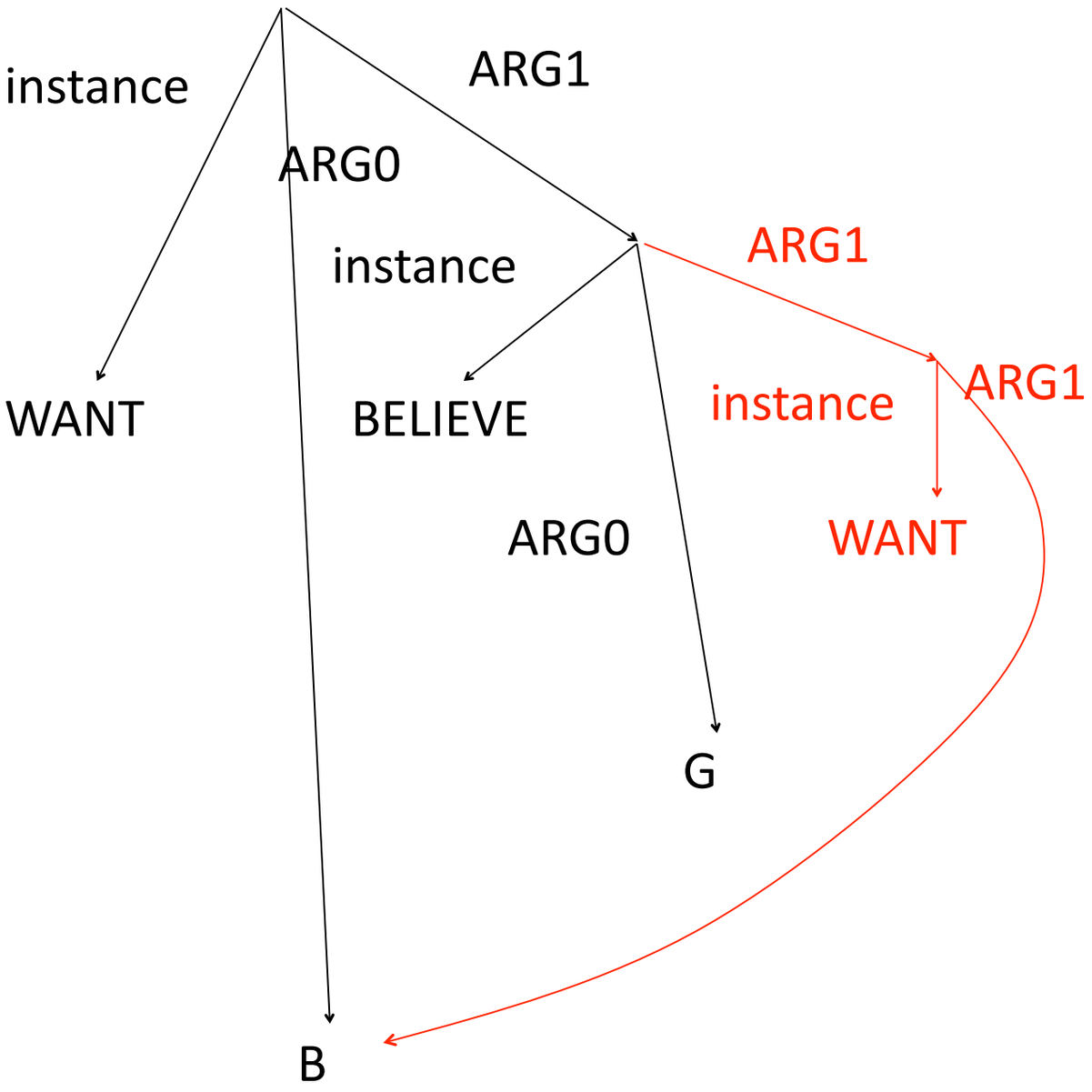


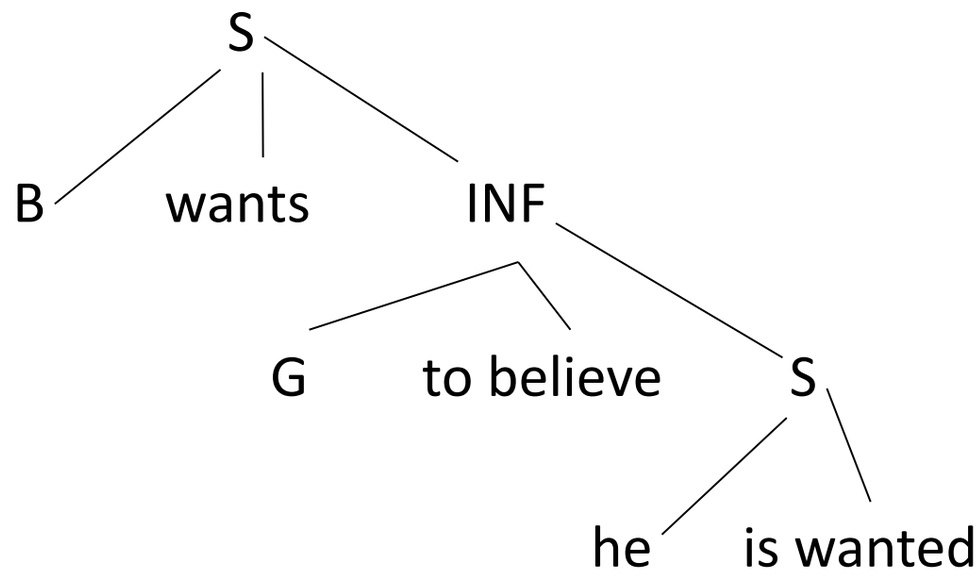
"the boy wants
something
involving himself"

SHRG Derivation

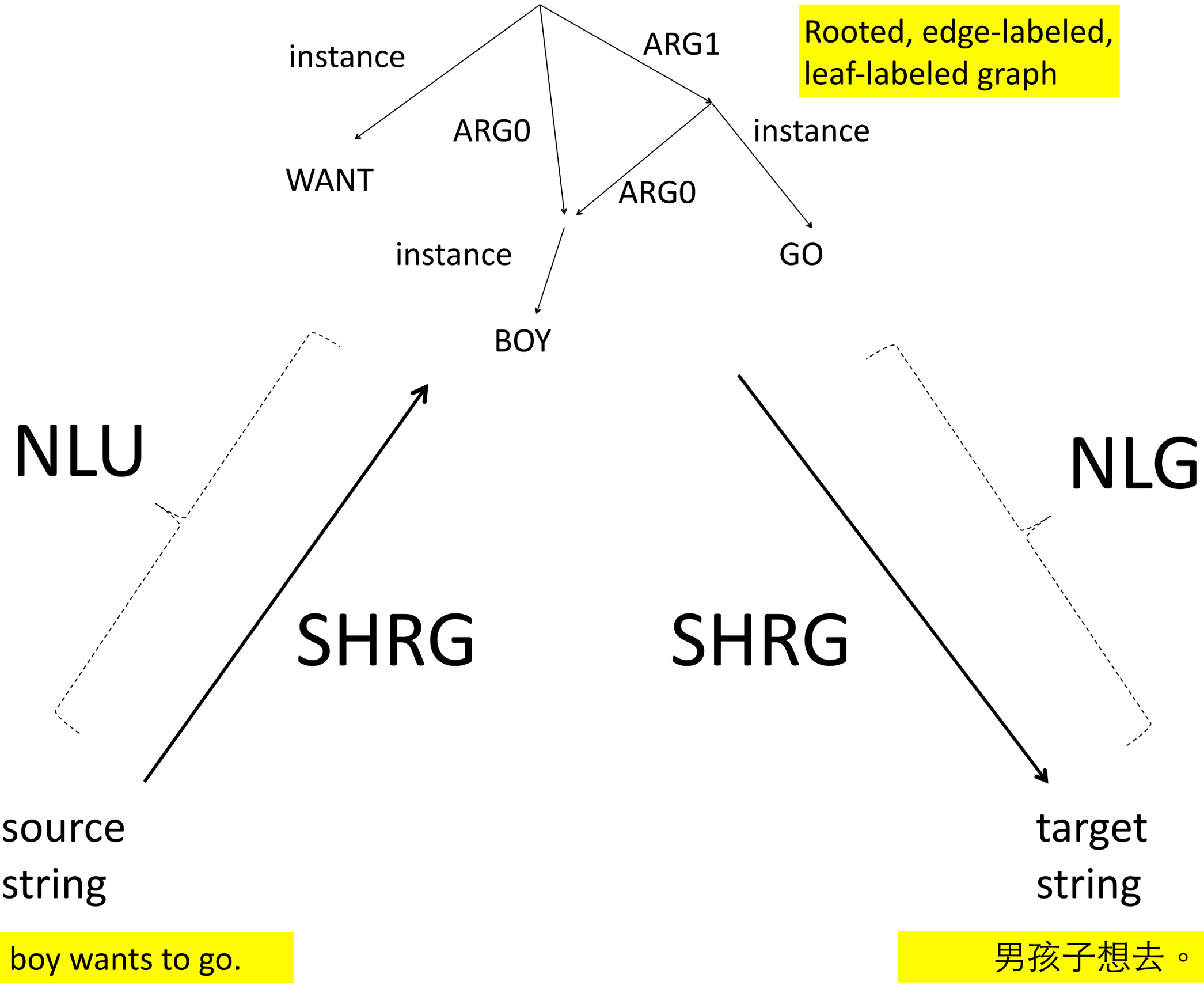


SHRG Derivation



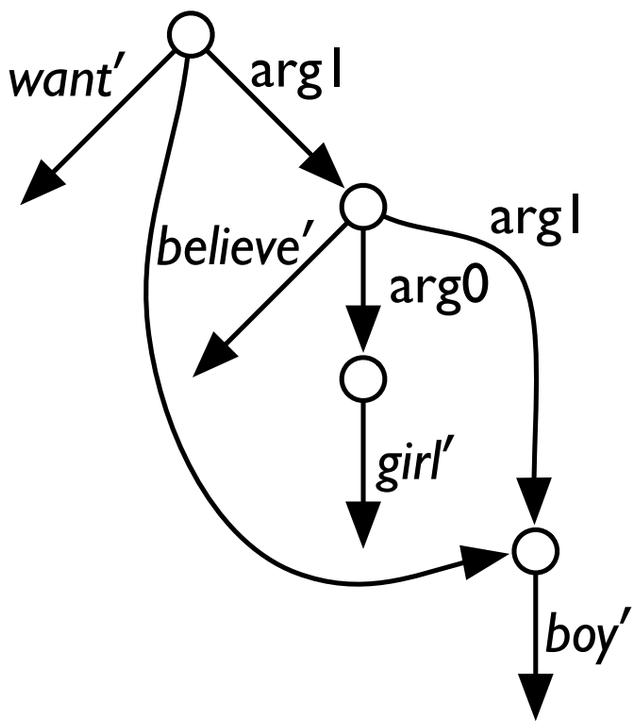


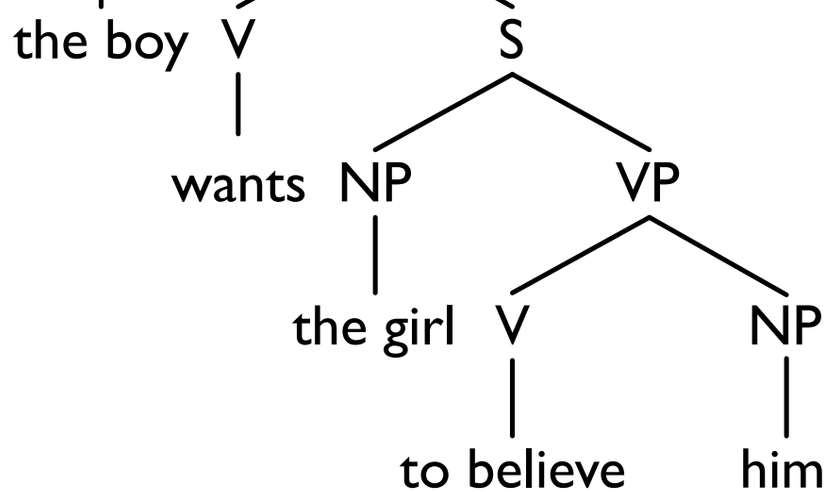
FINISHED!


instance
ARG1
Rooted, edge-labeled,
leaf-labeled graph
ARG0
instance
WANT
ARG0
instance
GO
BOY
NLU
NLG
SHRG
SHRG
source
string
target
string
The boy wants to go.
男孩子想去。

Recognizing (hyper)graphs

- Task: Given a graph *H*, find the best
  - derivation of *H*
  - transduction of *H* into a syntactic tree

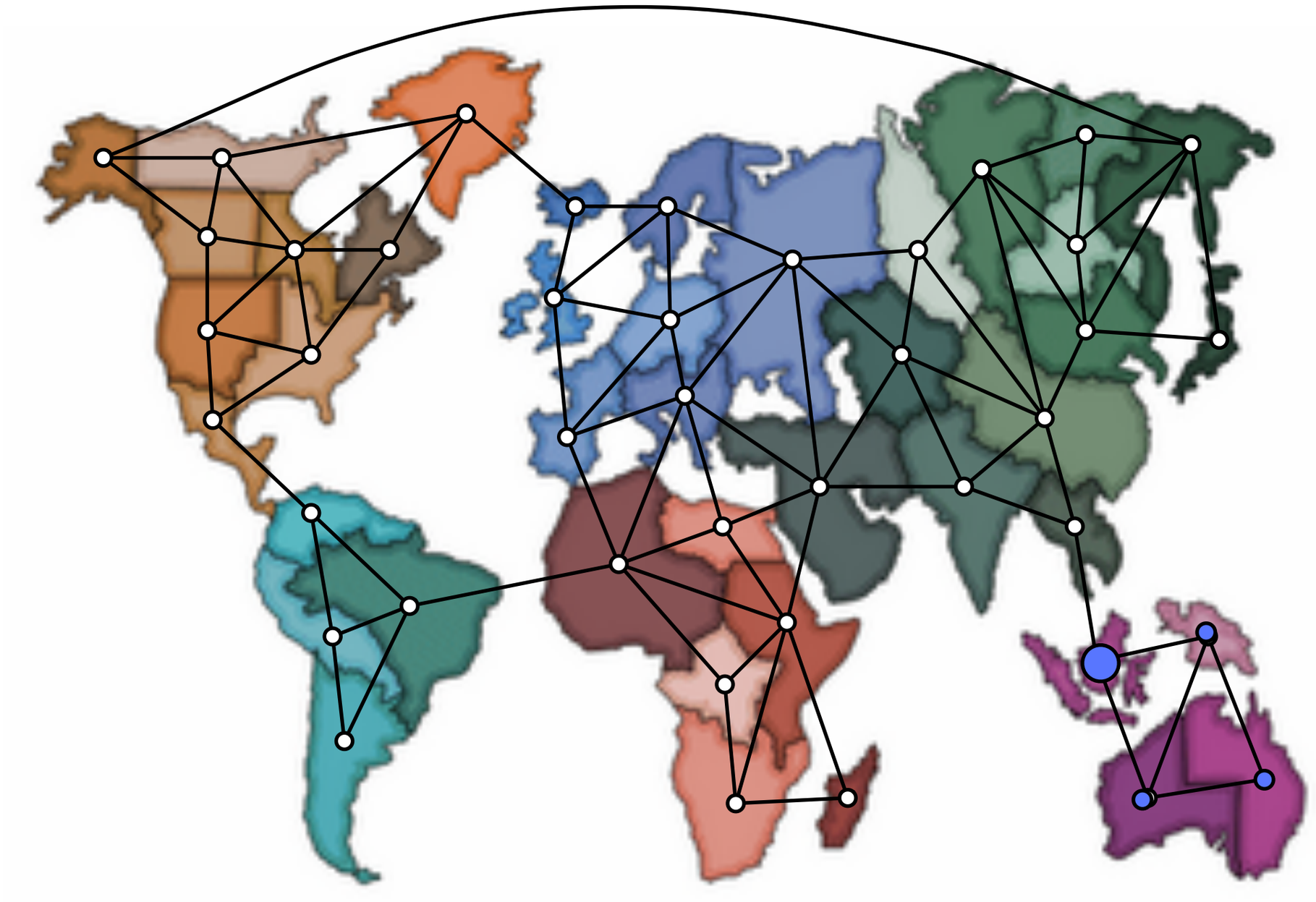




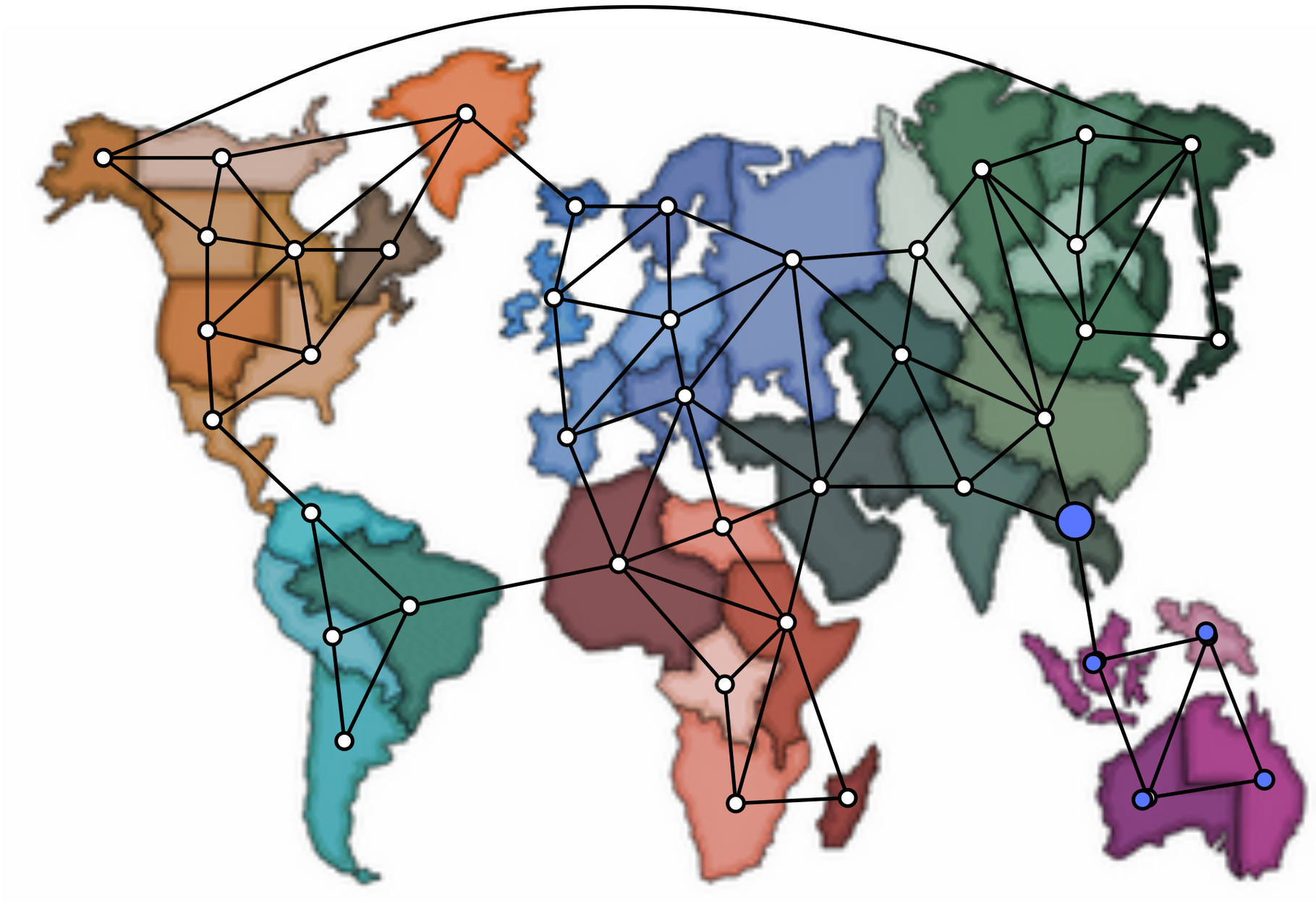
Recognizing (hyper)graphs

- Previous algorithms (Drewes 1997) fairly theoretical
- What we want:
  - Extract a bunch of HRG rules
  - Throw out the troublemakers
  - Guarantee recognition in $O(n^k)$ time

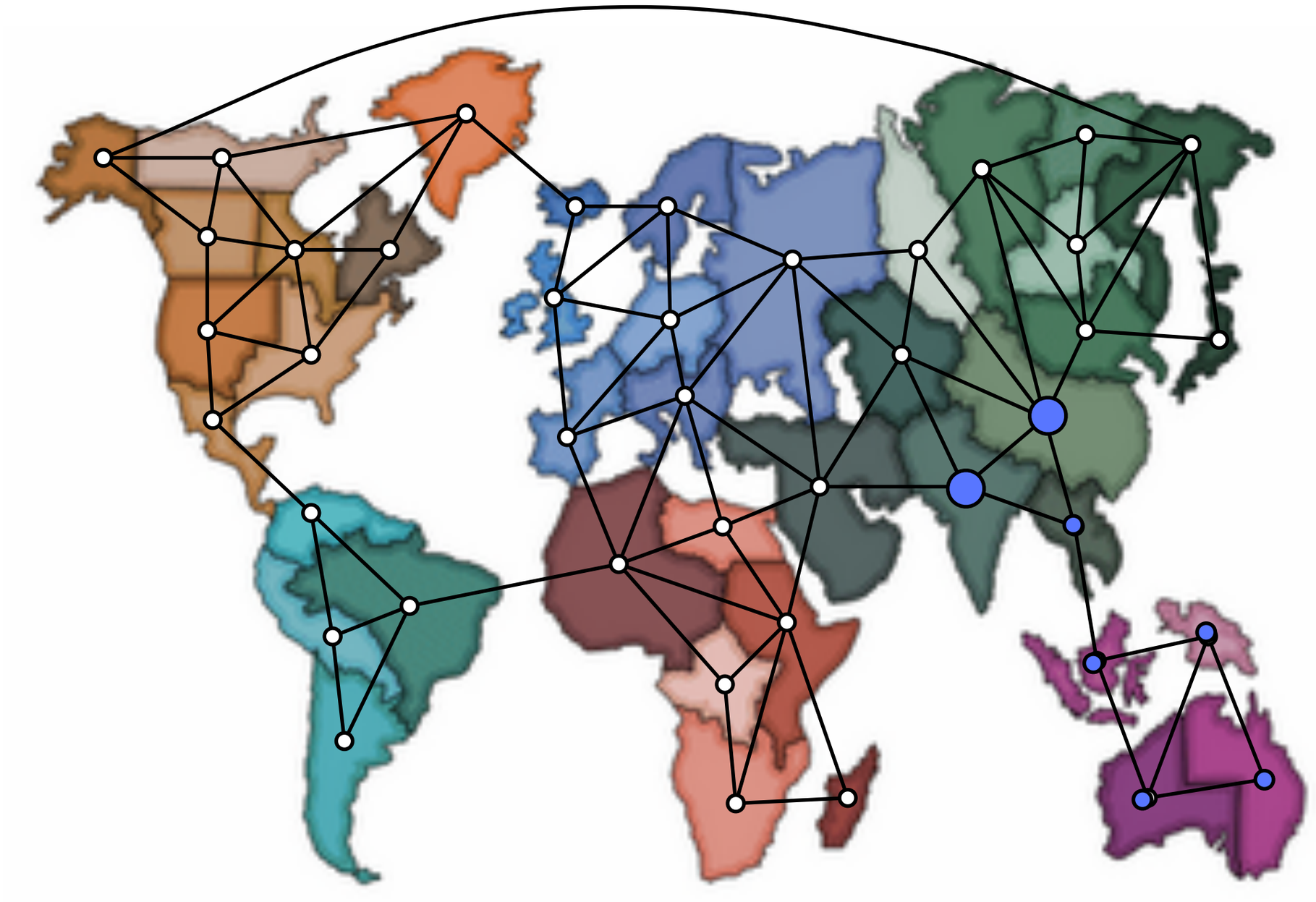
Pathwidth



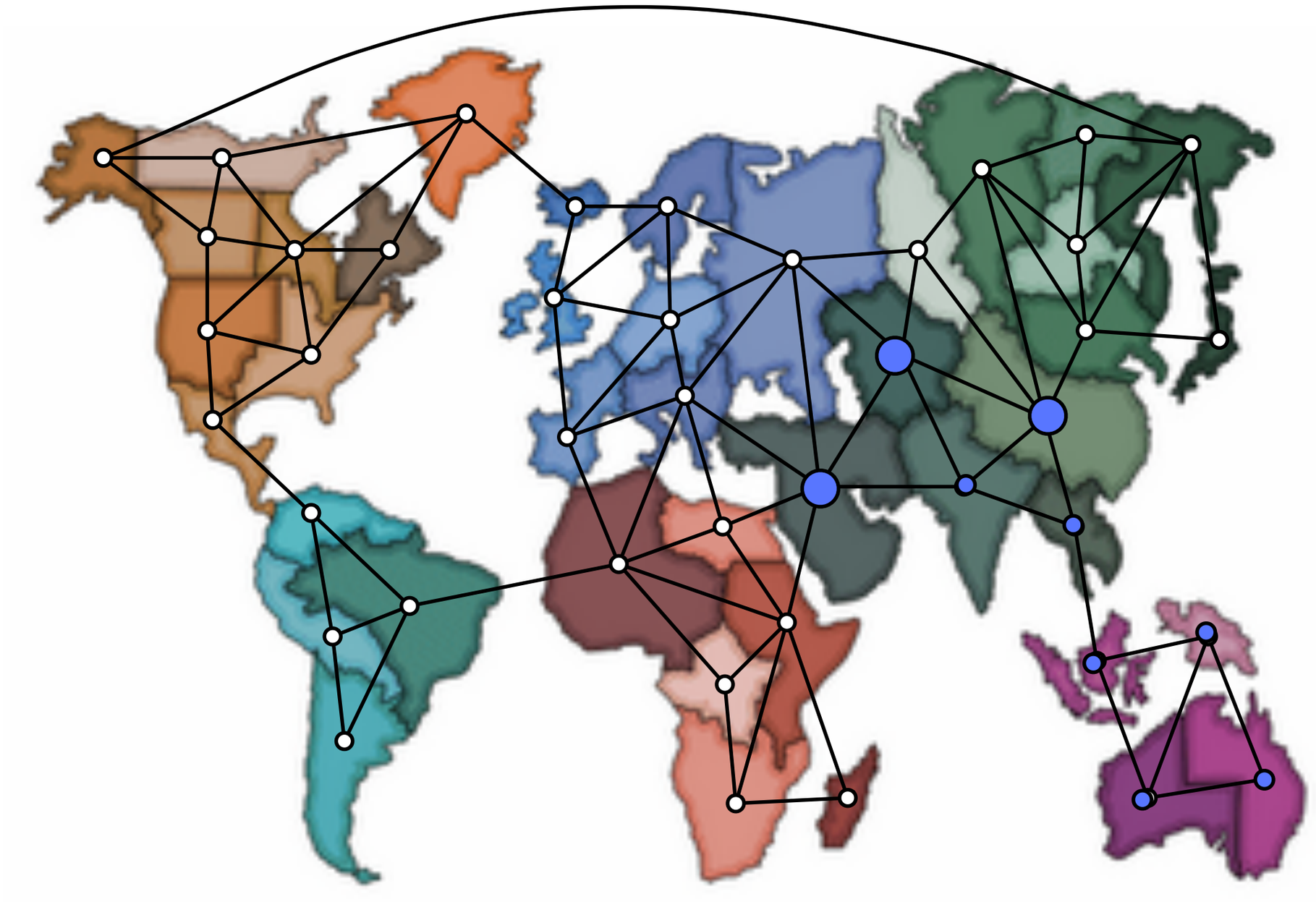
Pathwidth



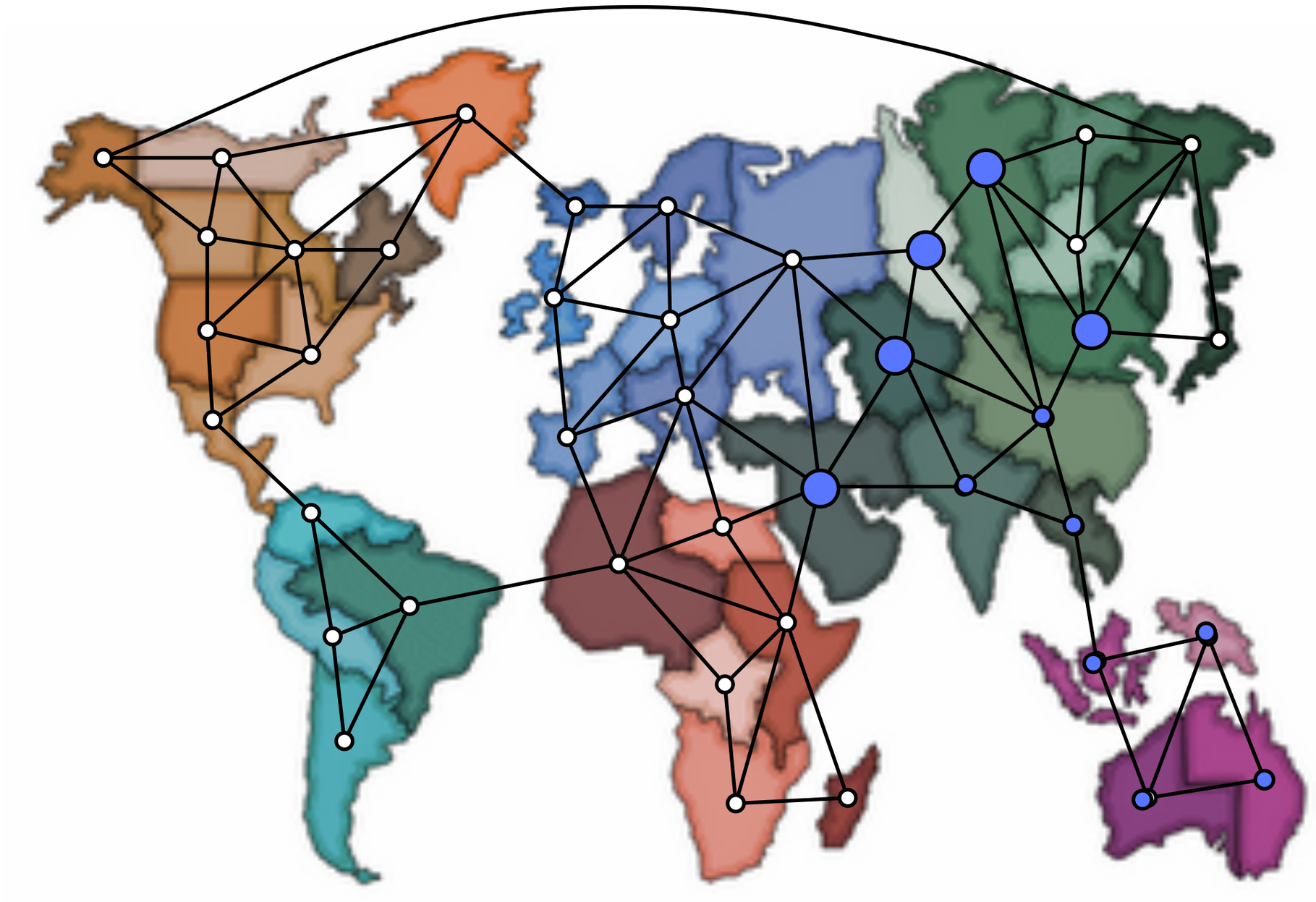
Pathwidth



Pathwidth



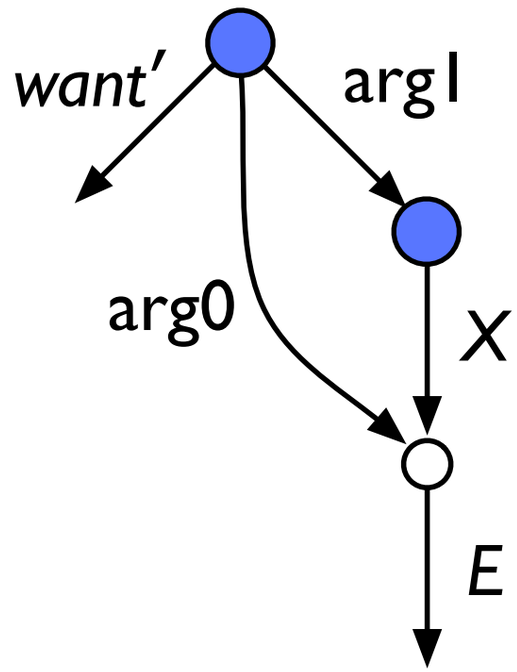
Pathwidth



Treewidth

- Pathwidth k = a single boundary with at most k nodes
- Treewidth k = multiple independent boundaries, each with at most k nodes

Recognizing (hyper)graphs



treewidth 2

- If rules have treewidth at most k, we can recognize in time $O(n^{k+1})$

Recognizing (hyper)graphs

treewidth	runtime	continents	strings (words)	trees (syntax)	graphs (semantics)
1	n^2			CFG	HRG in practice
2	n^3	Australia S America	CFG	TAG	
3	n^4	Africa			
4	n^5	N America Europe			
5	n^6	Asia	TAG		