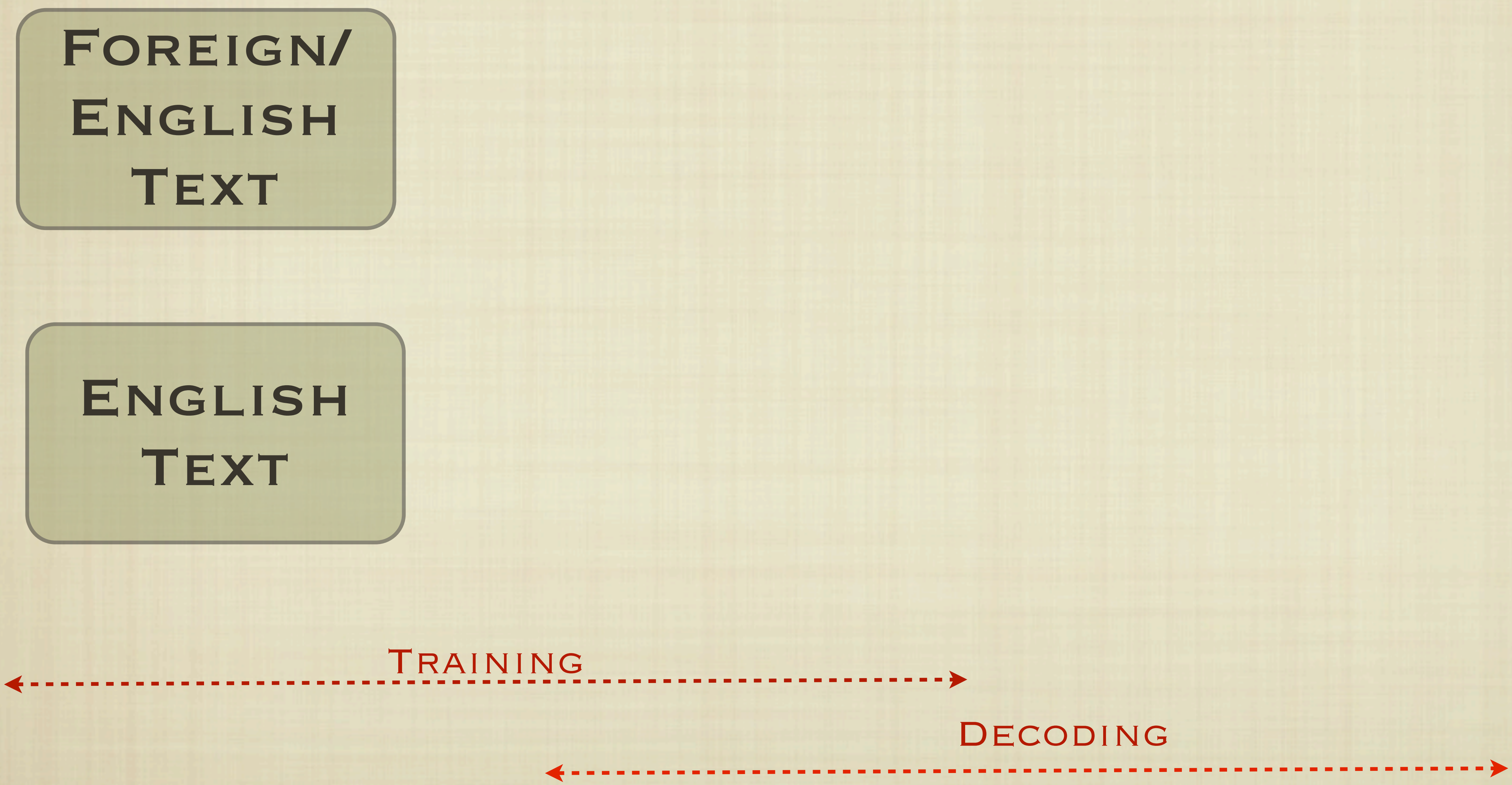
Machine Translation Decoding

Ahmad Emami
IBM T J Watson
emami@us.ibm.com

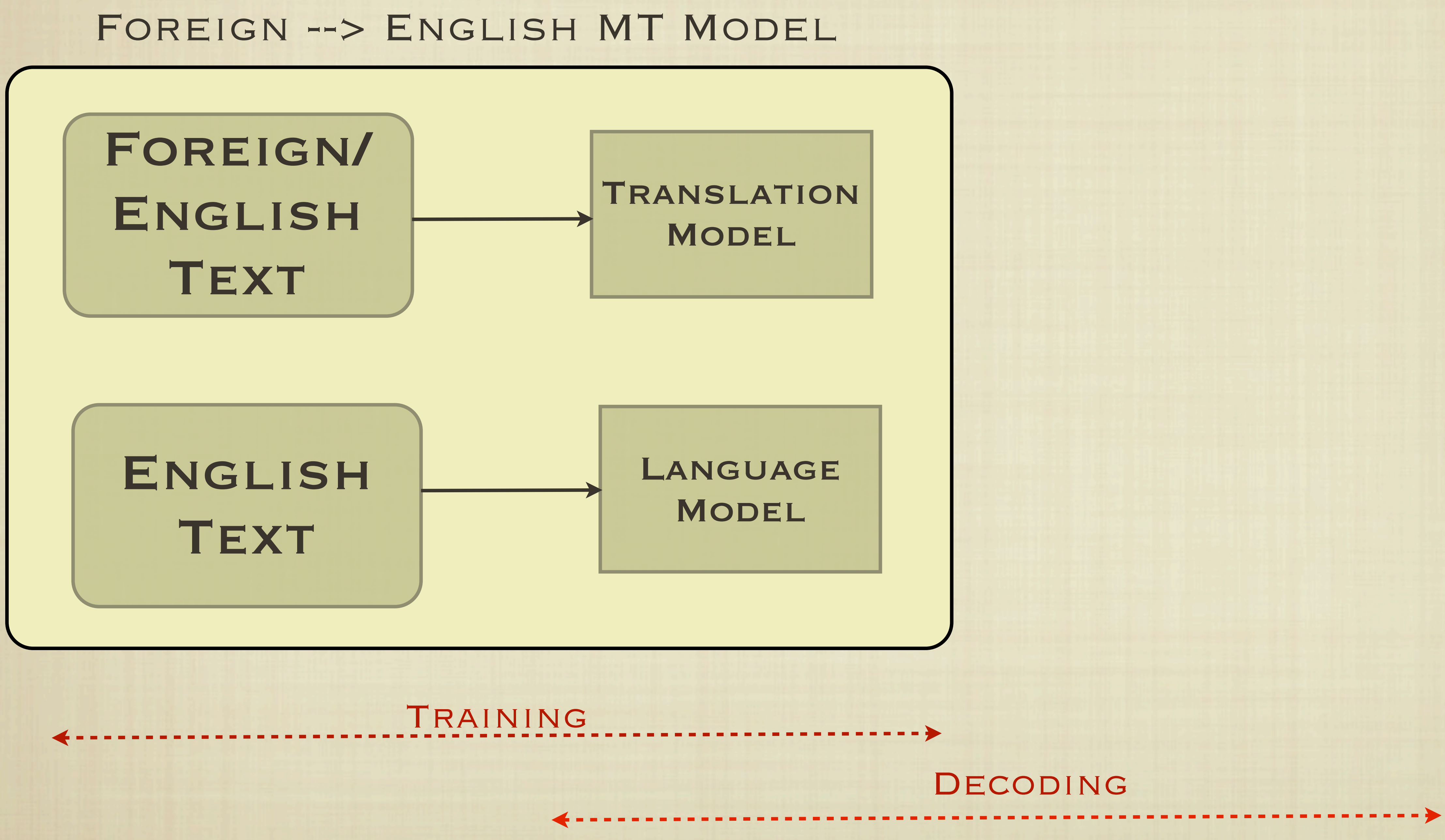
With slides from Philipp Koehn, David Chiang, and Yaser Al-Onaizan

Columbia University Nov 21, 2012

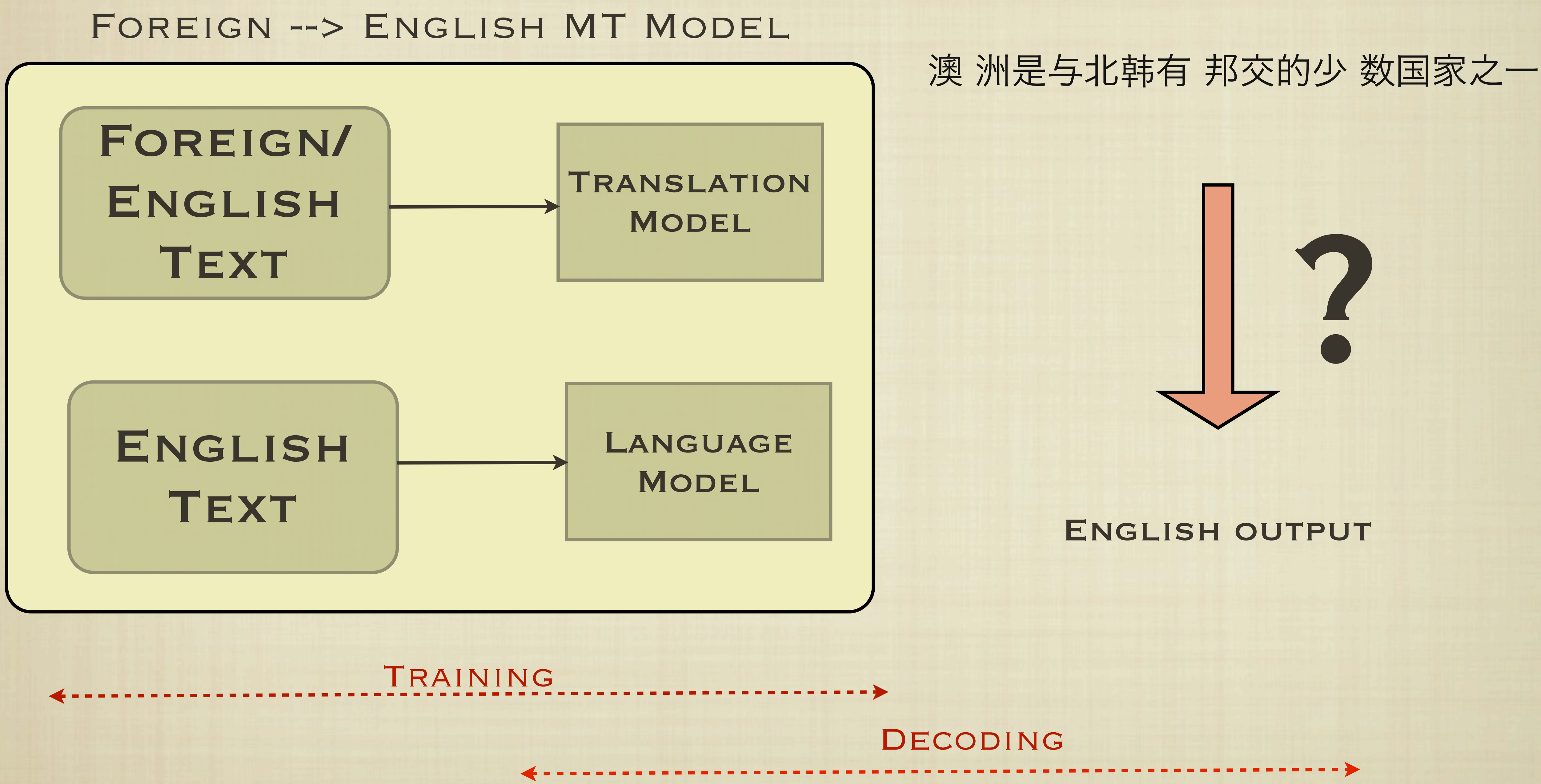
What we Have?

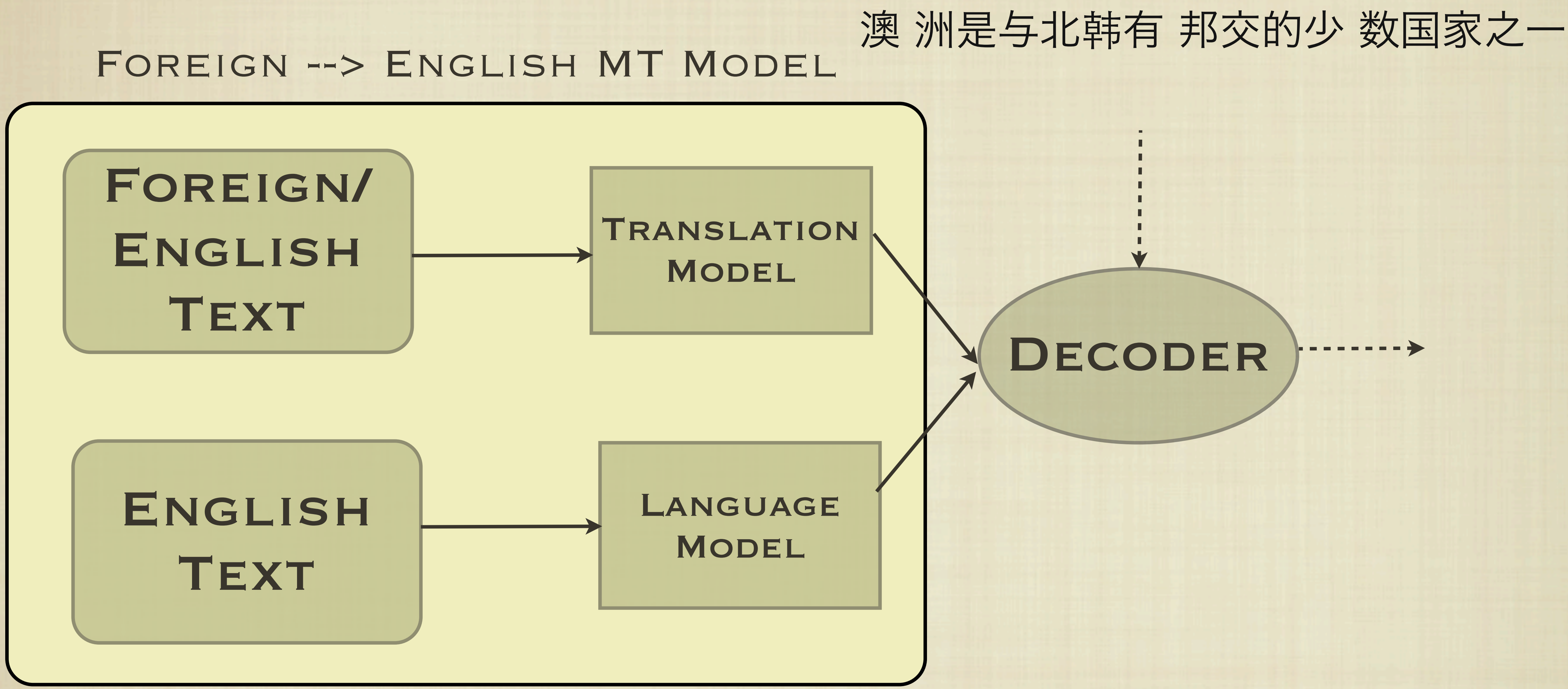


Translation and language models



Translation and language models




澳 洲是与北韩有 邦交的少 数国家之一
Foreign --> English MT Model
Foreign/
English
Text
Translation
Model
English
Text
Language
Model
Decoder

BRUTE FORCE SEARCH

- simple minded way to do translation is like what you are doing for homework 3
- find translations of each word with highest translation score and append them together
- List all possible translations, score them, and pick the best
- not good : why?

BRUTE FORCE SEARCH

A simple example:

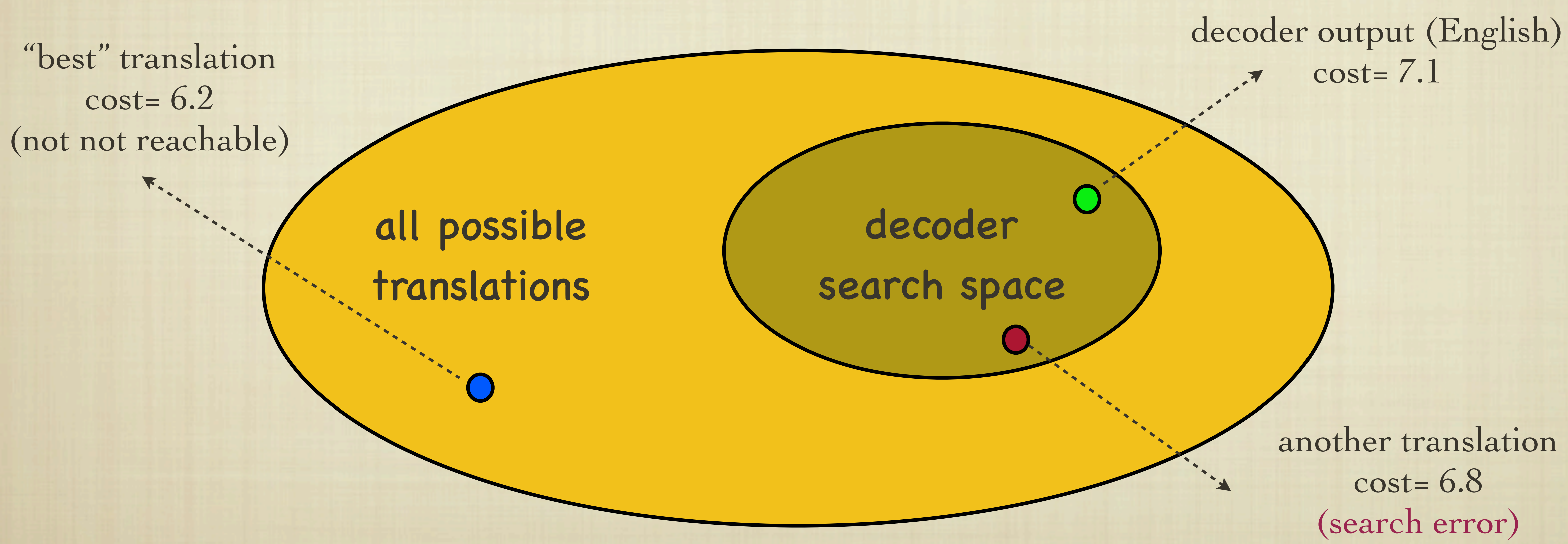
- 16 characters. Let's say each character has 5 possible translations
- 5^16= 152 billion possible translations
- Phrase-based translation --> different segmentation of the foreign sentence
- Re-ordering. A lot more combinations possible!

DECODER?

- We need a strategy to search among all possible translations in an efficient manner
- "Decoder" is the component that does the search

Decoder and search space

- Usually the search space is specified and limited by a grammar and/or the decoder itself
- Emphasis is on efficiency while making as few search errors as possible



General Search Strategy

- Create and Score "Partial Hypothesis"
- Each Partial Hyp is the translation of certain words of foreign language
- Group Partial Hyps
- In each group keep only the most promising Partial Hyps
- Extend Partial Hyps

Decoders:

- Phrase Decoder
- Hierarchical Decoder (Chart)
- Left-to-Right hierarchical
- Your own decoding strategy...

Phrase-based Decoder

- Other names: Beam Decoder, Stack Decoder, ...
- Left-to-right production of translation
- But can move around foreign sentence
- The moves are not totally arbitrary.
  Why?

Decoding Process

Maria	no	dio	una	bofetada	a	la	bruja	verde

- Build translation left to right
  - select foreign words to be translated

Decoding Process

- Build translation left to right
  - select foreign words to be translated
  - find English phrase translation
  - add English phrase to end of partial translation

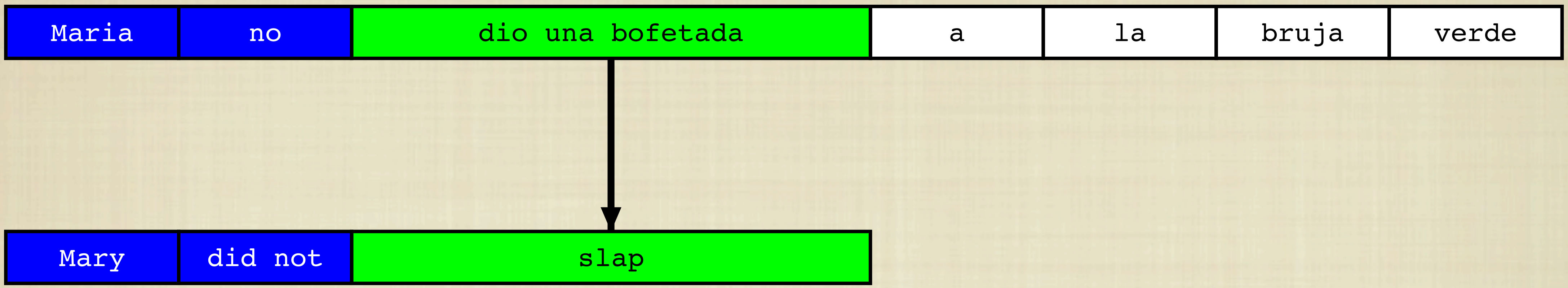
Decoding Process

- Build translation left to right
  - select foreign words to be translated
  - find English phrase translation
  - add English phrase to end of partial translation
  - mark foreign words as translated

Decoding Process

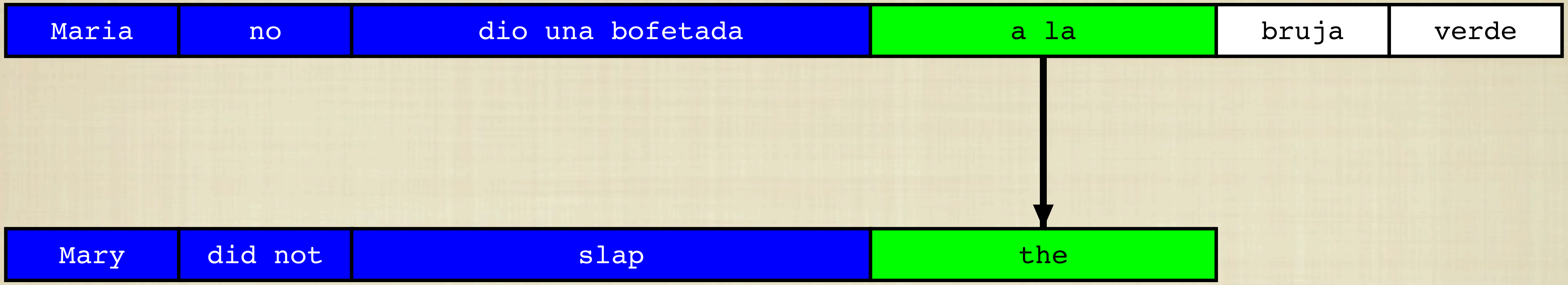
- One to many translation

Decoding Process



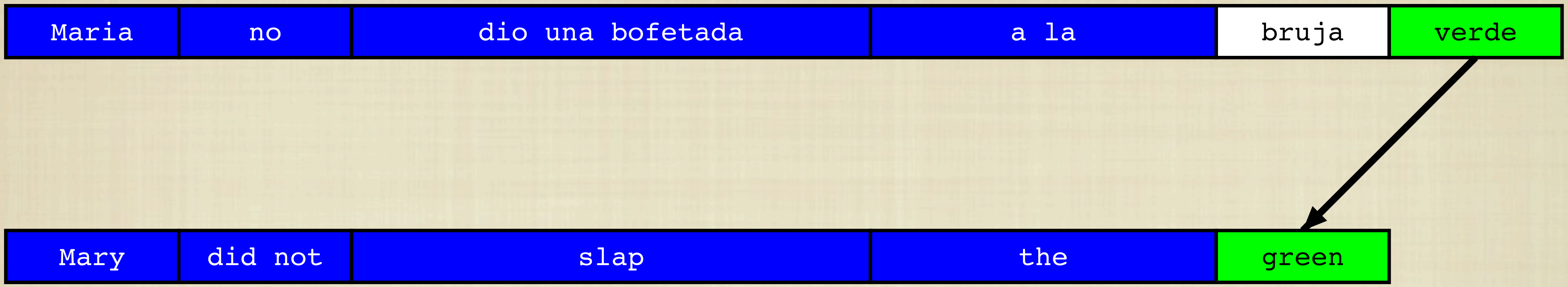
- Many to one translation

Decoding Process



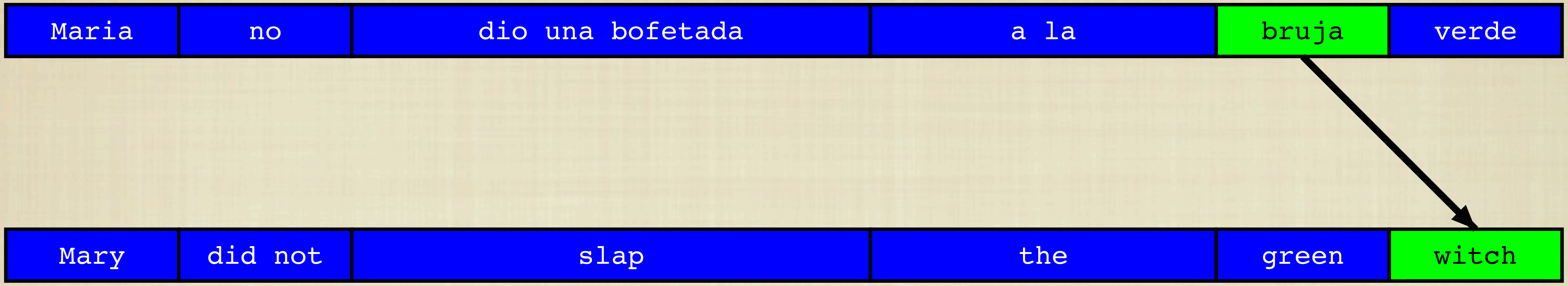
- Many to one translation

Decoding Process



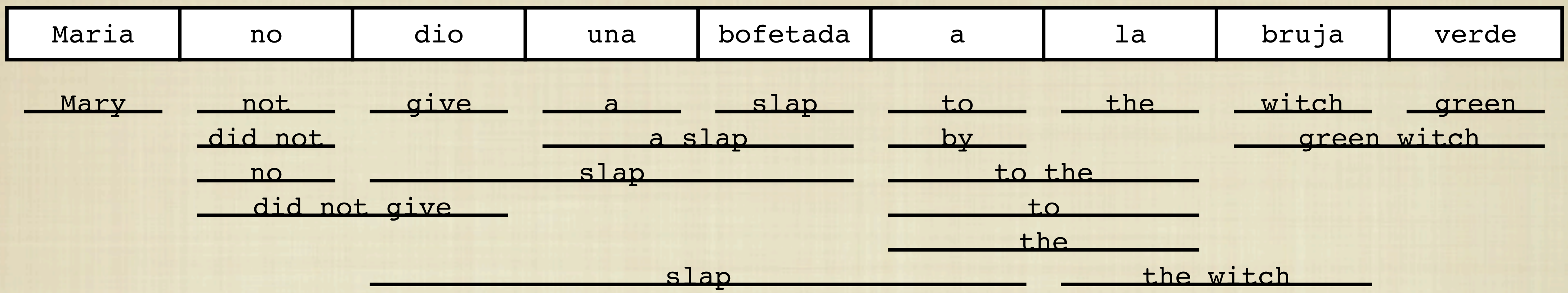
- Reordering

Decoding Process



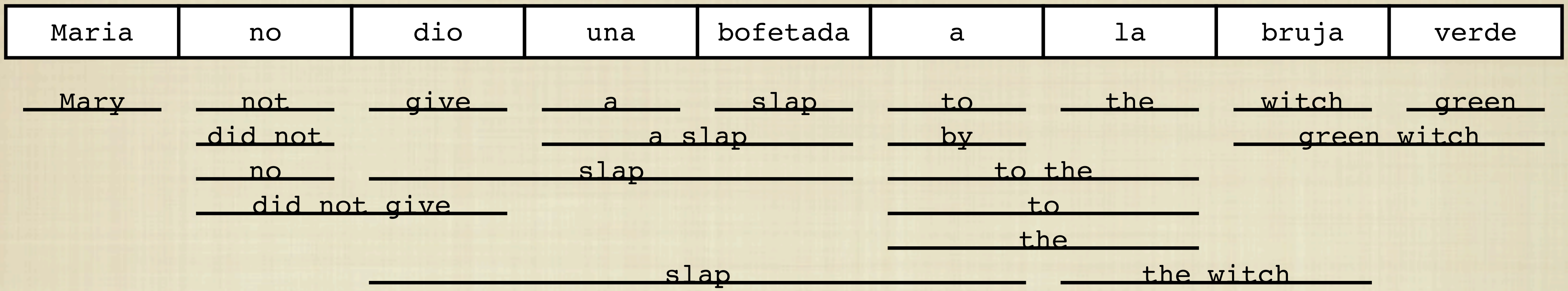
- Translation finished

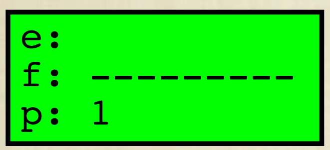
Translation Options



- Look up possible phrase translations
  - many different ways to segment words into phrases
  - many different ways to translate each phrase

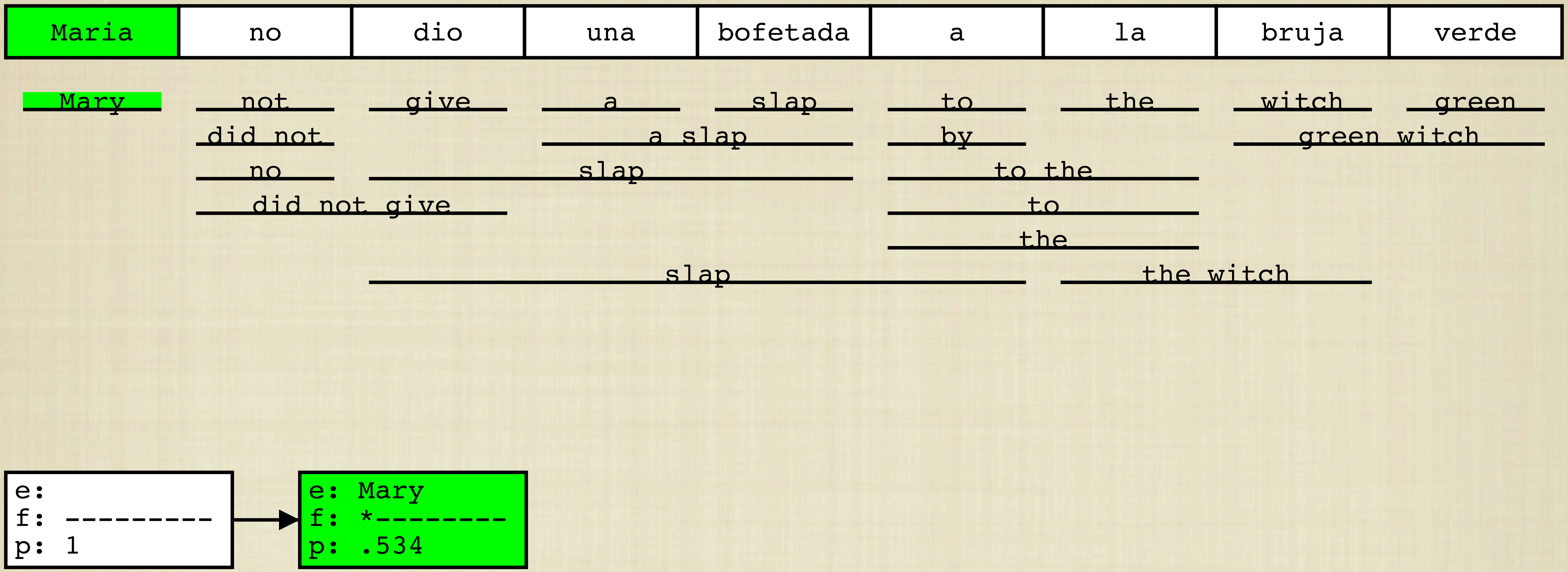
Hypothesis Expansion





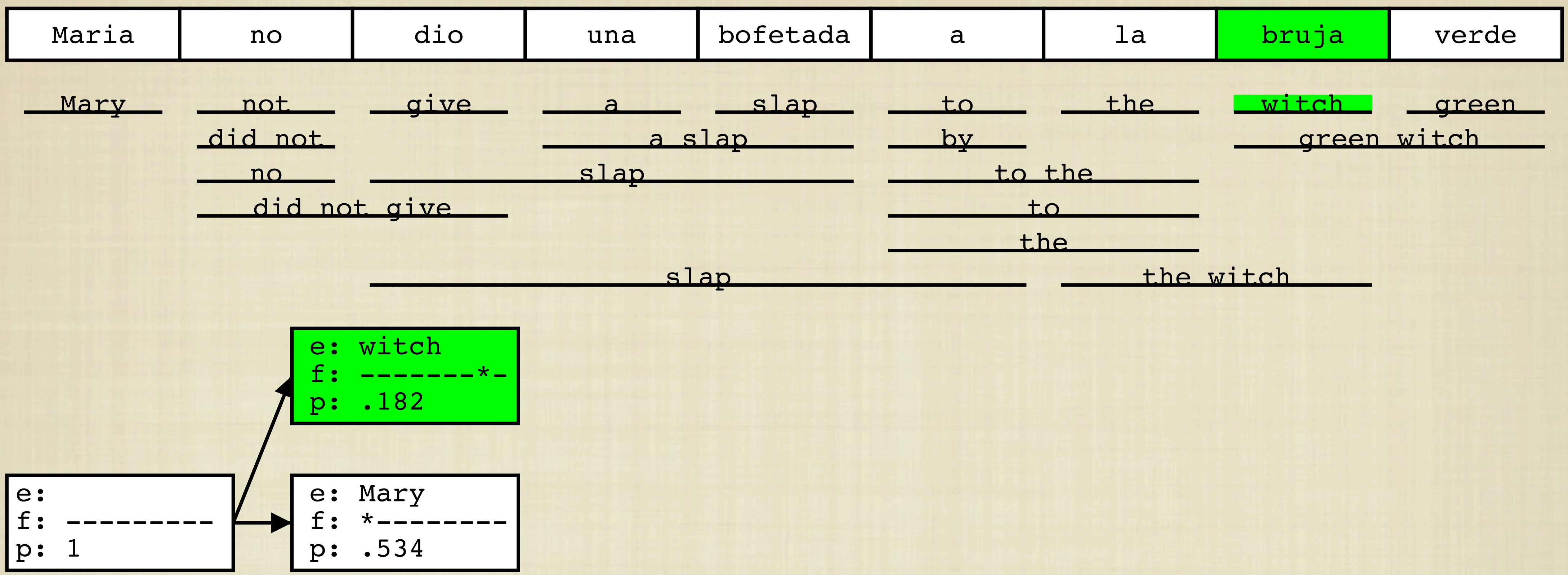
- Start with null hypothesis
  - e: no English words
  - f: no foreign words covered
  - p: probability 1

Hypothesis Expansion



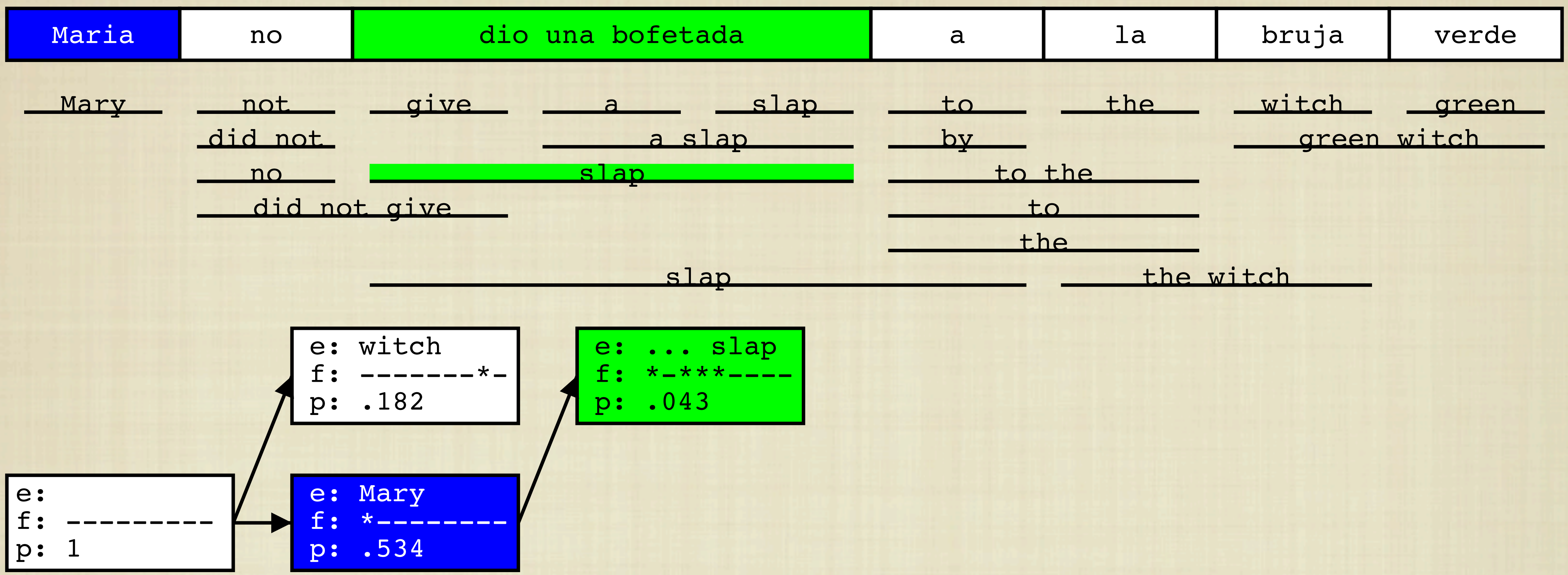
- Pick translation option
- Create hypothesis
  - e: add English phrase `Mary`
  - f: first foreign word covered
  - p: probability 0.534

Hypothesis Expansion



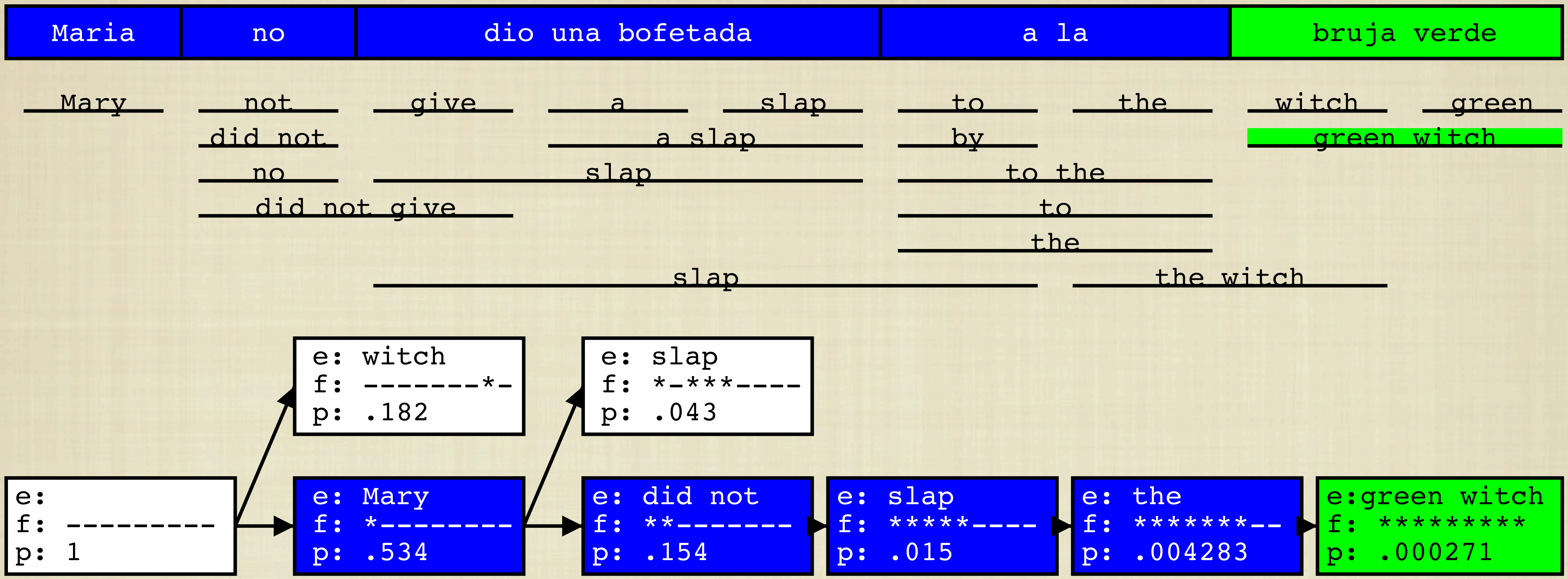
- Add another hypothesis

Hypothesis Expansion



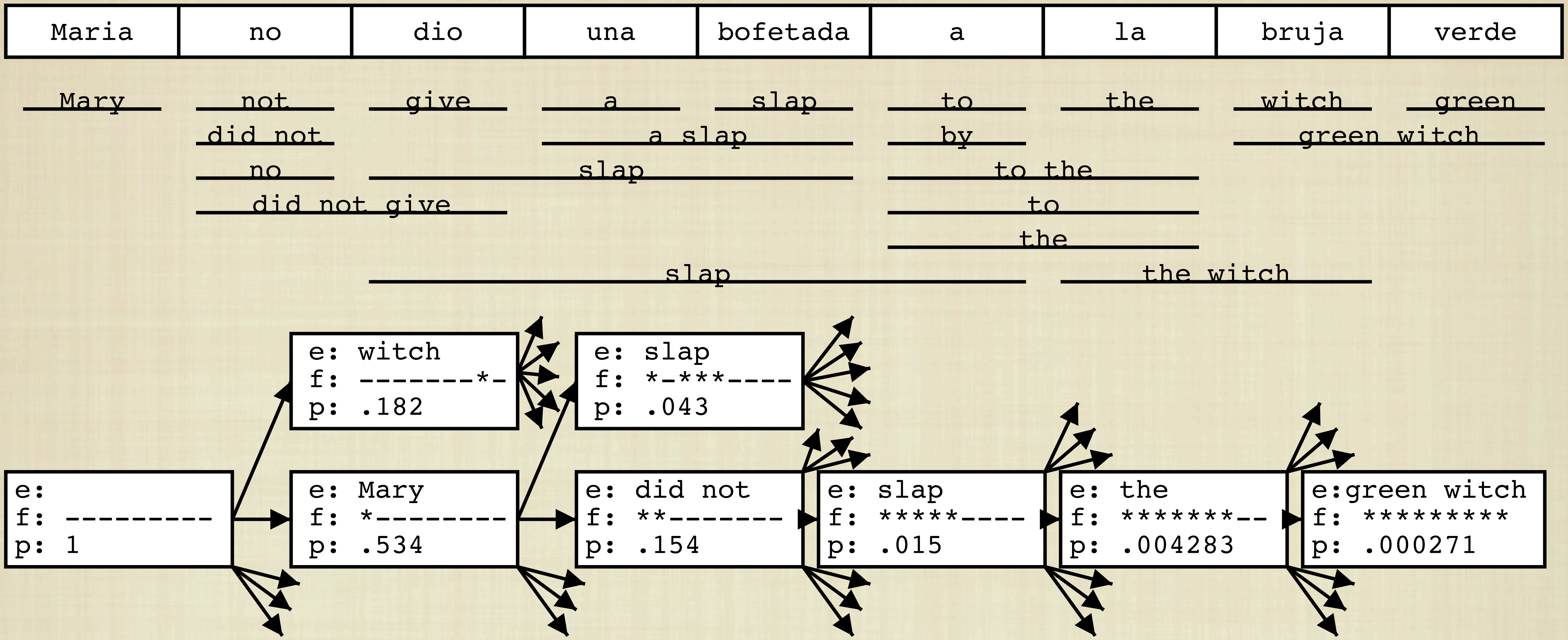
- Further hypothesis expansion

Hypothesis Expansion



- ... until all foreign words covered
  - find best hypothesis that covers all foreign words
  - backtrack to read off translation

Hypothesis Expansion



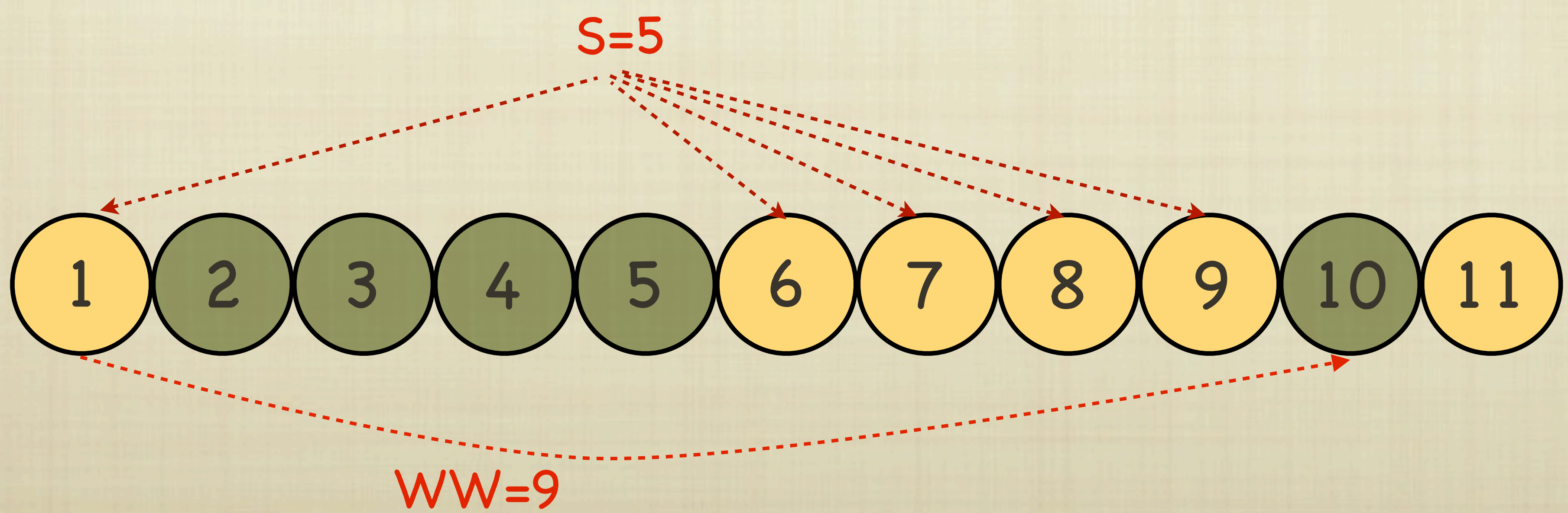
- Adding more hypothesis

⇒ Explosion of search space

Restricting the search space

- Search space is too big.
- Use the two following restrictions on movements:

1. **Skip**: how many source words can we temporarily skip during translation
2. **Window Width**: how far to the right can we go before we are forced to translate a skipped word



Explosion of Search Space

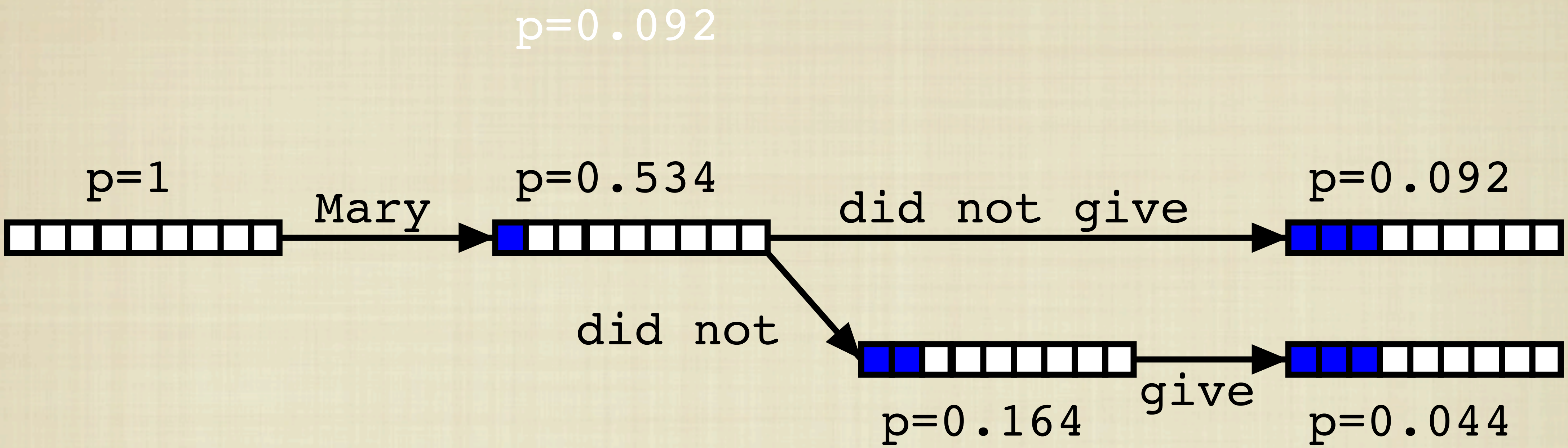
- Number of hypotheses is exponential with respect to sentence length

⇒ Decoding is NP-complete [Knight, 1999]

⇒ Need to reduce search space

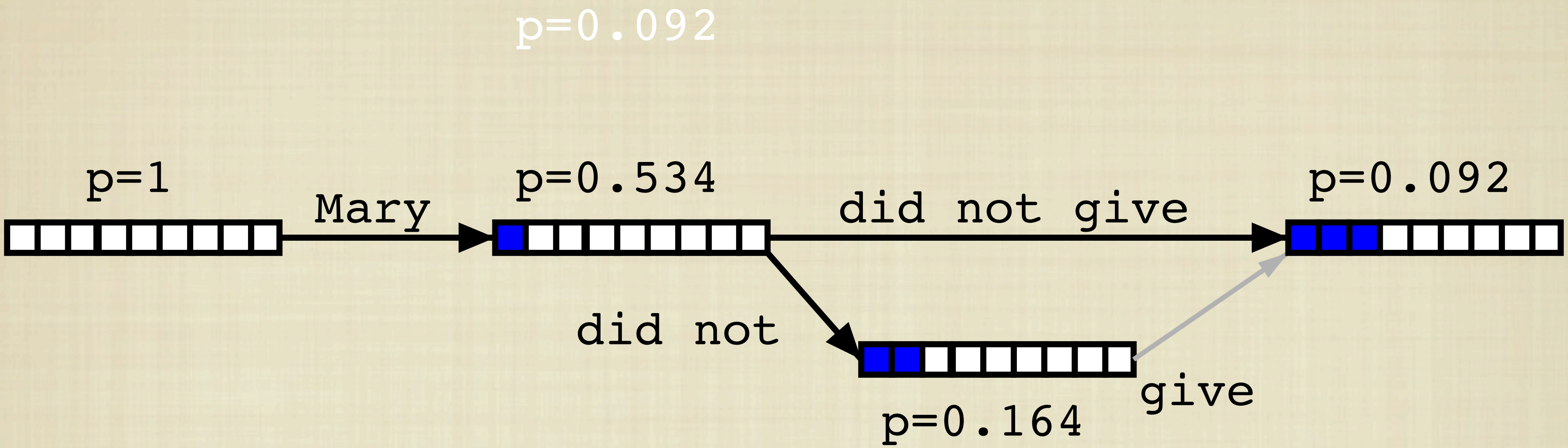
  - risk free: hypothesis recombination
  - risky: histogram/threshold pruning

Hypothesis Recombination



- Different paths to the same partial translation

Hypothesis Recombination

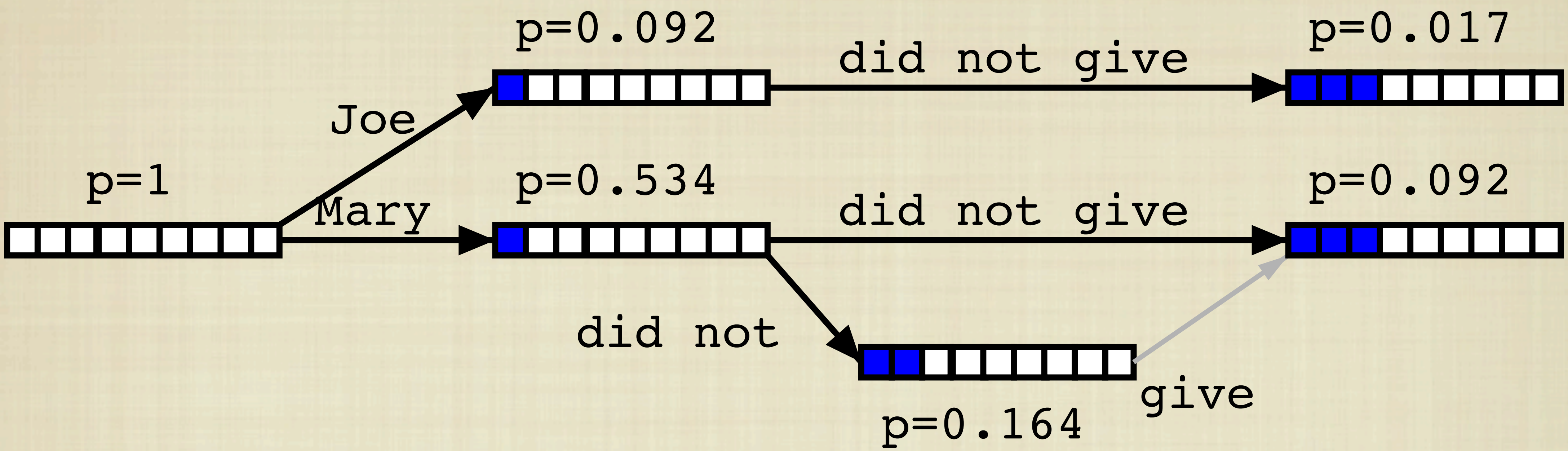


- Different paths to the same partial translation

⇒ Combine paths

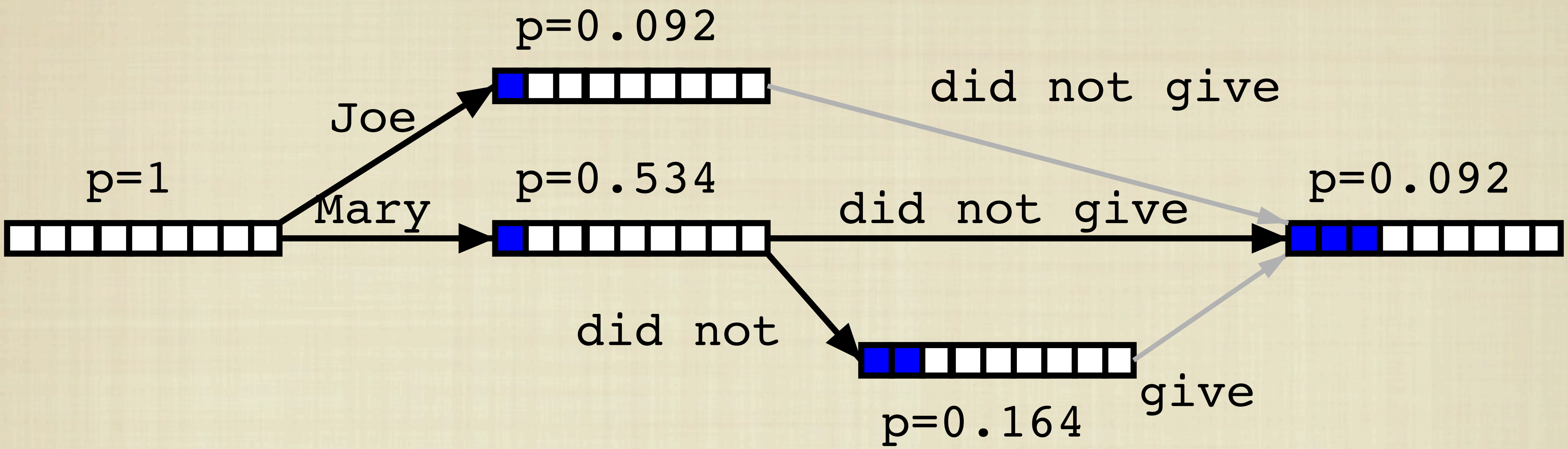
  - drop weaker hypothesis
  - keep pointer from worse path

Hypothesis Recombination



- Recombined hypotheses do not have to match completely
- No matter what is added, weaker path can be dropped, if:
  - last two English words match (matters for language model)
  - foreign word coverage vectors match (effects future path)

Hypothesis Recombination



- Recombined hypotheses do not have to match completely
- No matter what is added, weaker path can be dropped, if:
  - last two English words match (matters for language model)
  - foreign word coverage vectors match (effects future path)

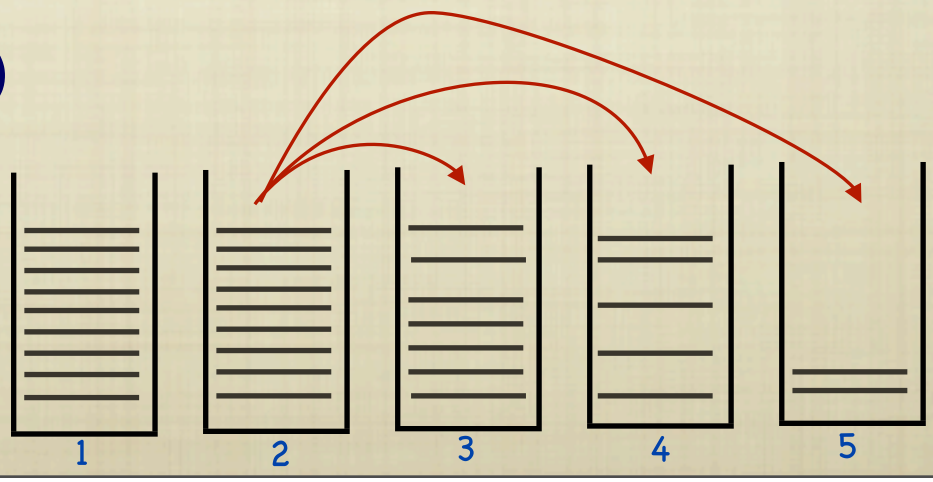
⇒ Combine paths

Pruning

- Hypothesis recombination is not sufficient

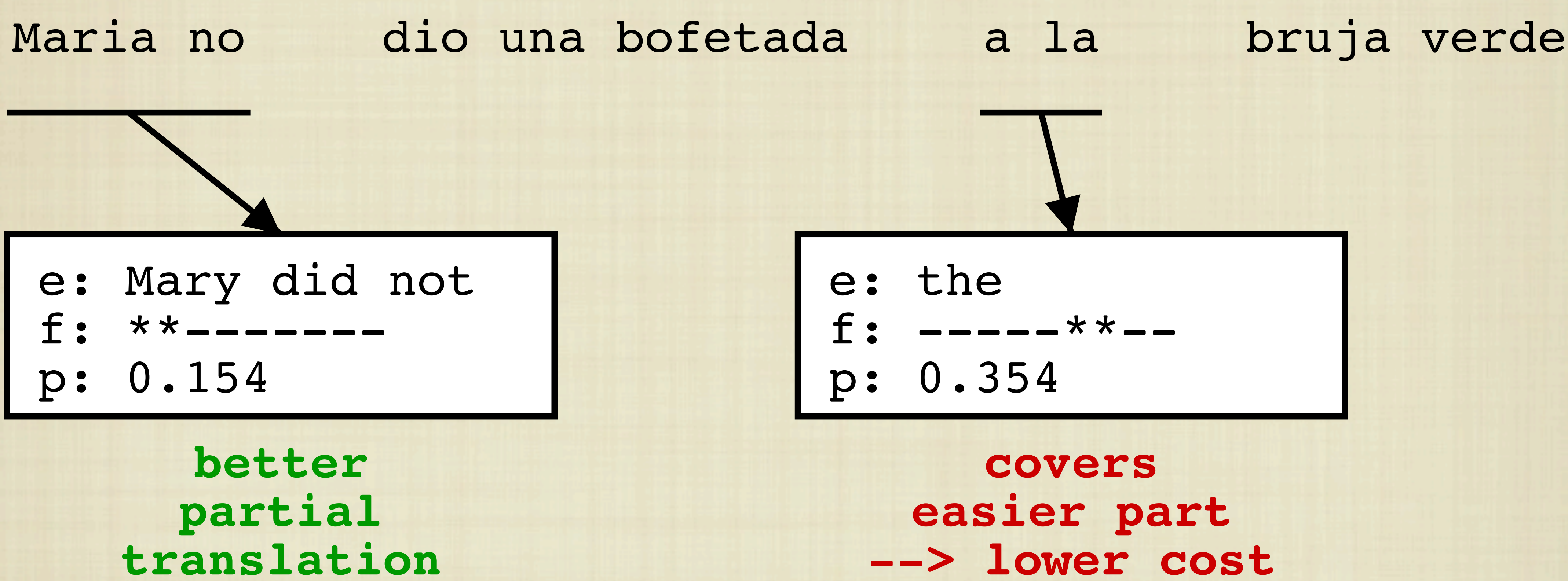
$\Rightarrow$ Heuristically discard weak hypotheses

- Organize Hypothesis in stacks, e.g. by
  - same foreign words covered
  - same number of foreign words covered (Pharaoh does this)
  - same number of English words produced
- Compare hypotheses in stacks, discard bad ones
  - histogram pruning: keep top n hypotheses in each stack (e.g., n=100)
  - threshold pruning: keep hypotheses that are at most α times the cost of best hypothesis in stack (e.g., α = 0.001)



Comparing Hypotheses

- Comparing hypotheses with same number of foreign words covered



- Hypothesis that covers *easy* part of sentence is preferred

$\Rightarrow$ Need to consider future cost

Future Cost Estimation

- Estimate cost to translate remaining part of input
- Step 1: find cheapest translation options
  - find cheapest translation option for each input span
  - compute translation model cost
  - estimate language model cost (no prior context)
  - ignore reordering model cost
- Step 2: compute cheapest cost
  - for each contiguous span:
  - find cheapest sequence of translation options
- Precompute and lookup
  - precompute future cost for each contiguous span
  - future cost for any coverage vector:
    sum of cost of each contiguous span of uncovered words

  $\rightarrow$ no expensive computation during run time

Outline

- Phrase-Based Statistical MT
- Beam Search Decoding
- **Experiments**
- Advanced Features

Experiments

- Decoder has to be evaluated in terms of search errors
  - translation errors not due to search errors are a challenge to the translation model
  - do not rely on search errors for good translation quality!
- Experimental setup
  - German to English
  - Europarl training corpus (30 million words)
  - 1500 sentence test corpus (avg. length 28.9 words)
  - 3 Ghz Linux machine, needs 512 MB RAM
  - Focus: illustrate trade-off speed / search errors
- Not measuring true search error
  - it is not tractable to find truly best translation
  - → relative to best translation found with high beam and different settings

Threshold Pruning

Threshold	0.0001	0.001	0.01	0.05	0.08
Time per Sentence	149 sec	119 sec	70 sec	27 sec	18 sec
Search Errors	-	+0%	+0%	+0%	+0%
Threshold	0.1	0.15	0.2	0.3	
Time per Sentence	15 sec	13 sec	10 sec	7 sec	
Search Errors	+1%	+3%	+6%	+12%	

- Low ratio of search errors for threshold $\alpha \leq 0.1$
- Results depend on weights for models

Histogram Pruning

Beam Size	1000	200	100	50	20	10	5
Time	15s	15s	14s	10s	9s	9s	7s
Search Errors	+1%	+1%	+2%	+4%	+8%	+20%	+35 %

- Low ratio of search errors for beam size $n \geq 200$

Translation Table Entries per Input Phrase

T-Table Limit	1000	500	200	100	50	20	10	5
Time	15.0s	7.6s	3.8s	1.9s	0.9s	0.4s	0.2s	0.1s
Search Errors	+1%	+1%	+1%	+1%	+1%	+2%	+7%	+18%

- Low ratio of search errors for limit of ≥ 50 entries in the translation table for each source language phrase
- About 1 second per sentence (30 words per second)
- Your mileage may vary

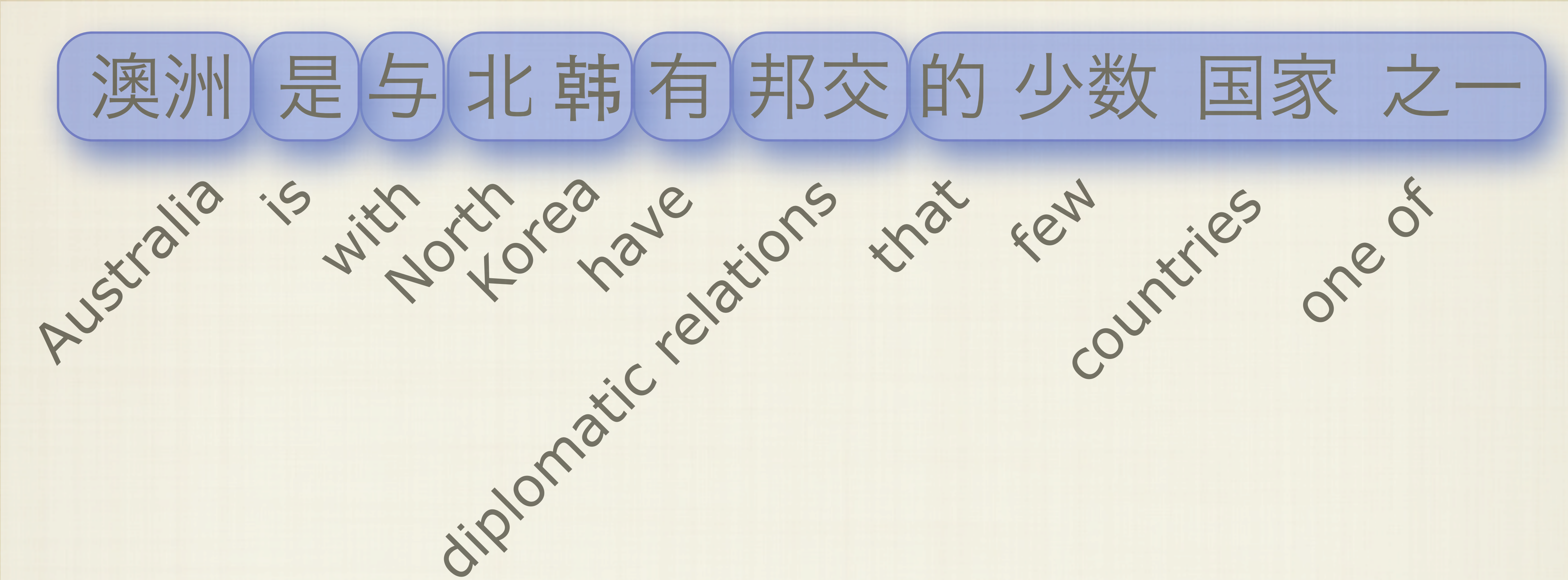
Hierarchical Decoder

Phrase-Based Decoder

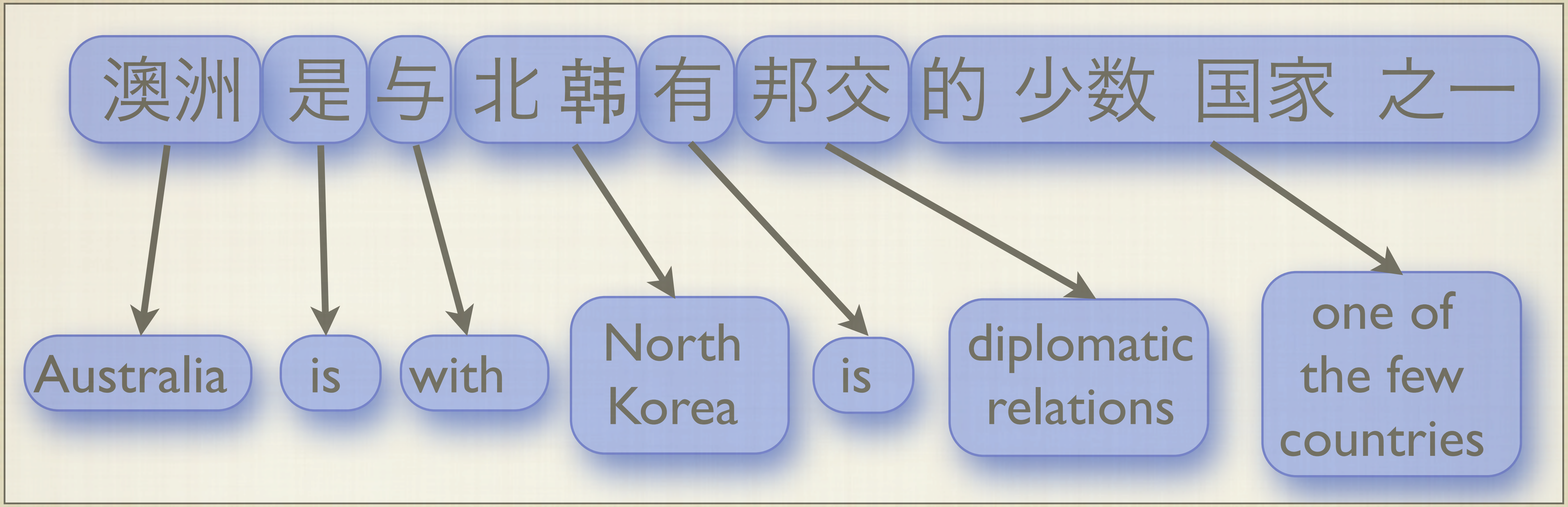
澳洲 是 与 北 韩 有 邦交 的 少数 国家 之一

Australia is with North Korea have diplomatic relations that few countries one of

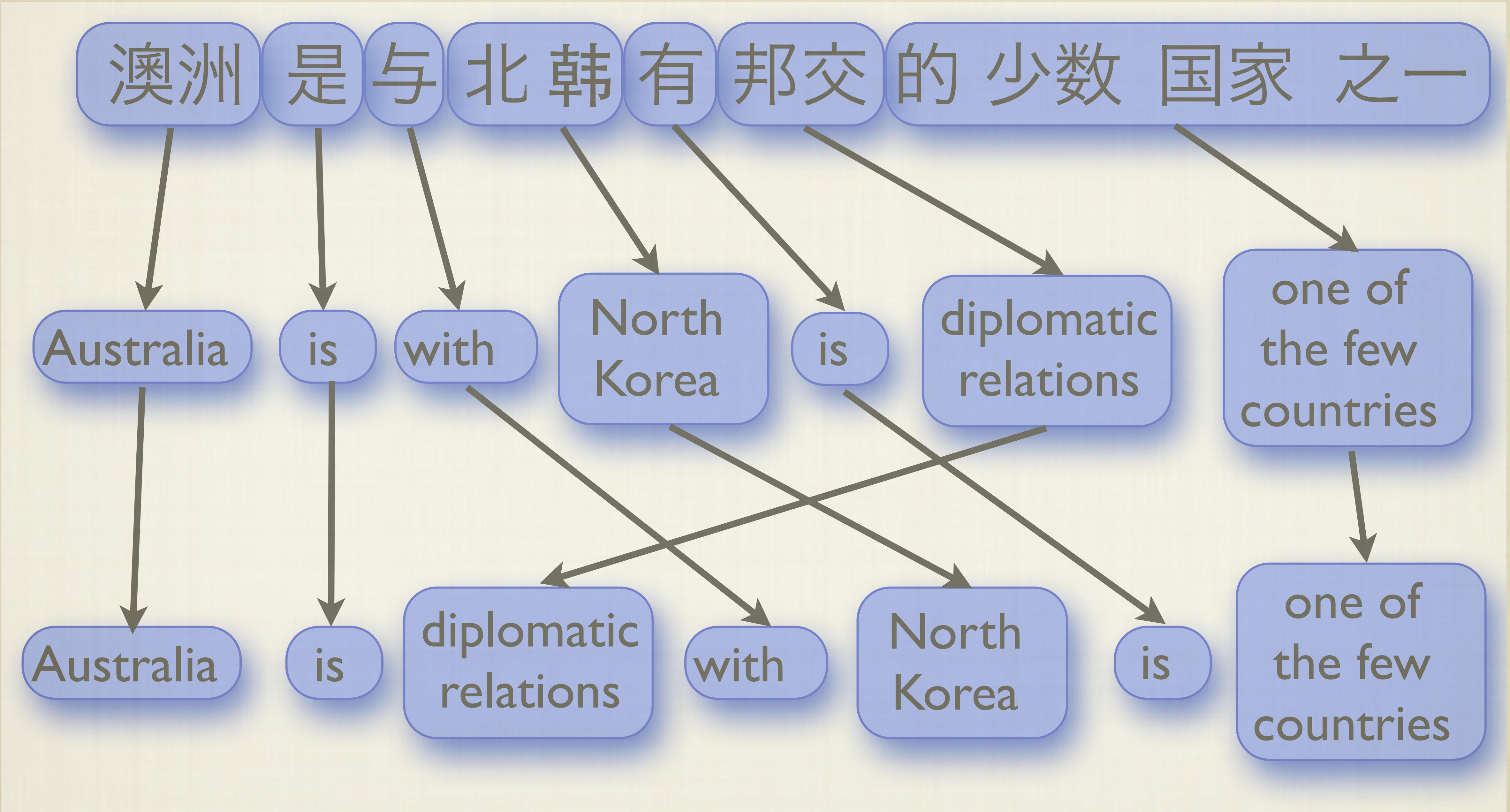
Phrase-Based Decoder



Phrase-Based Decoder



Phrase-Based Decoder



what's wrong here?

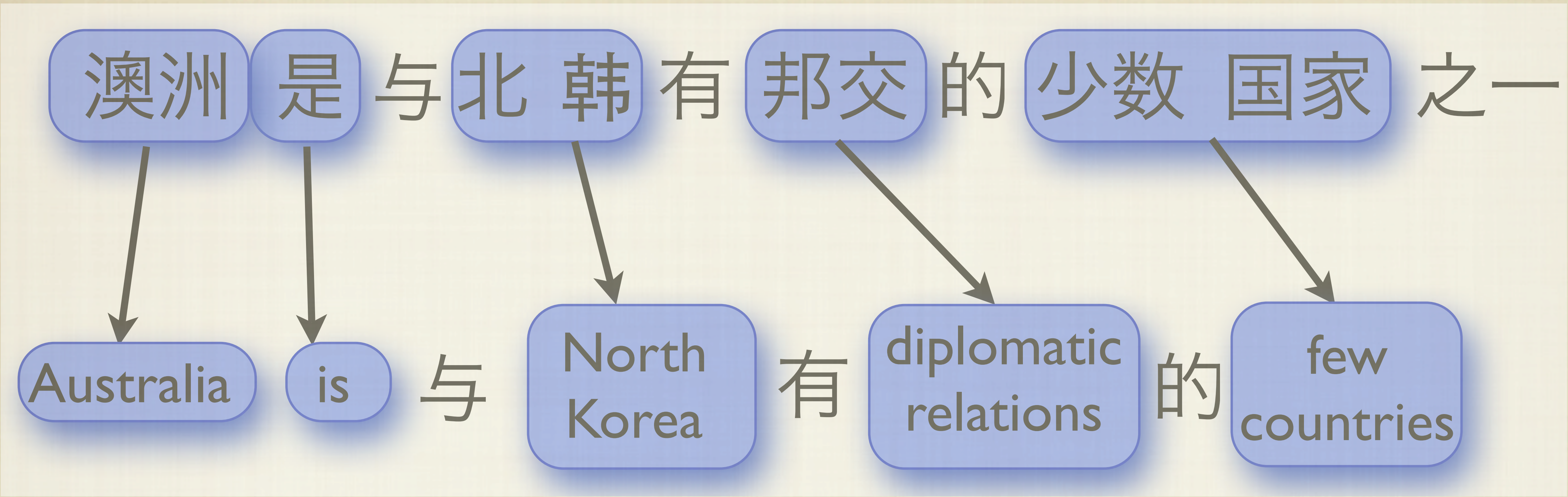
Phrase-based vs. Hierarchical

- Phrase-Based decoder is good at capturing local re-orderings
- Hierarchical decoding can capture re-orderings among phrases themselves
  - Useful for capturing long-distance movements

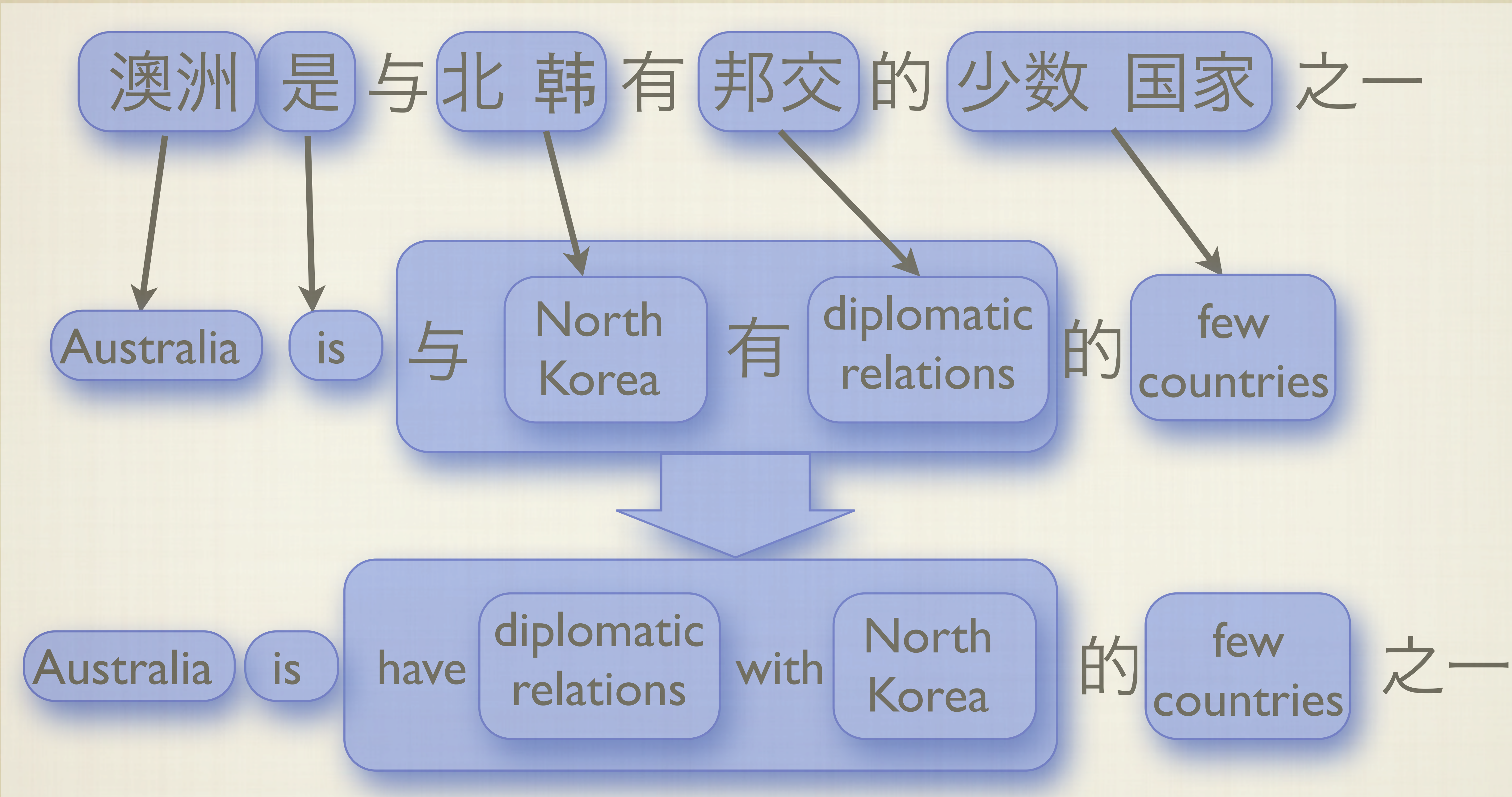
Hierarchical Translation

澳洲 是 与北 韩 有 邦交 的 少数 国家 之一

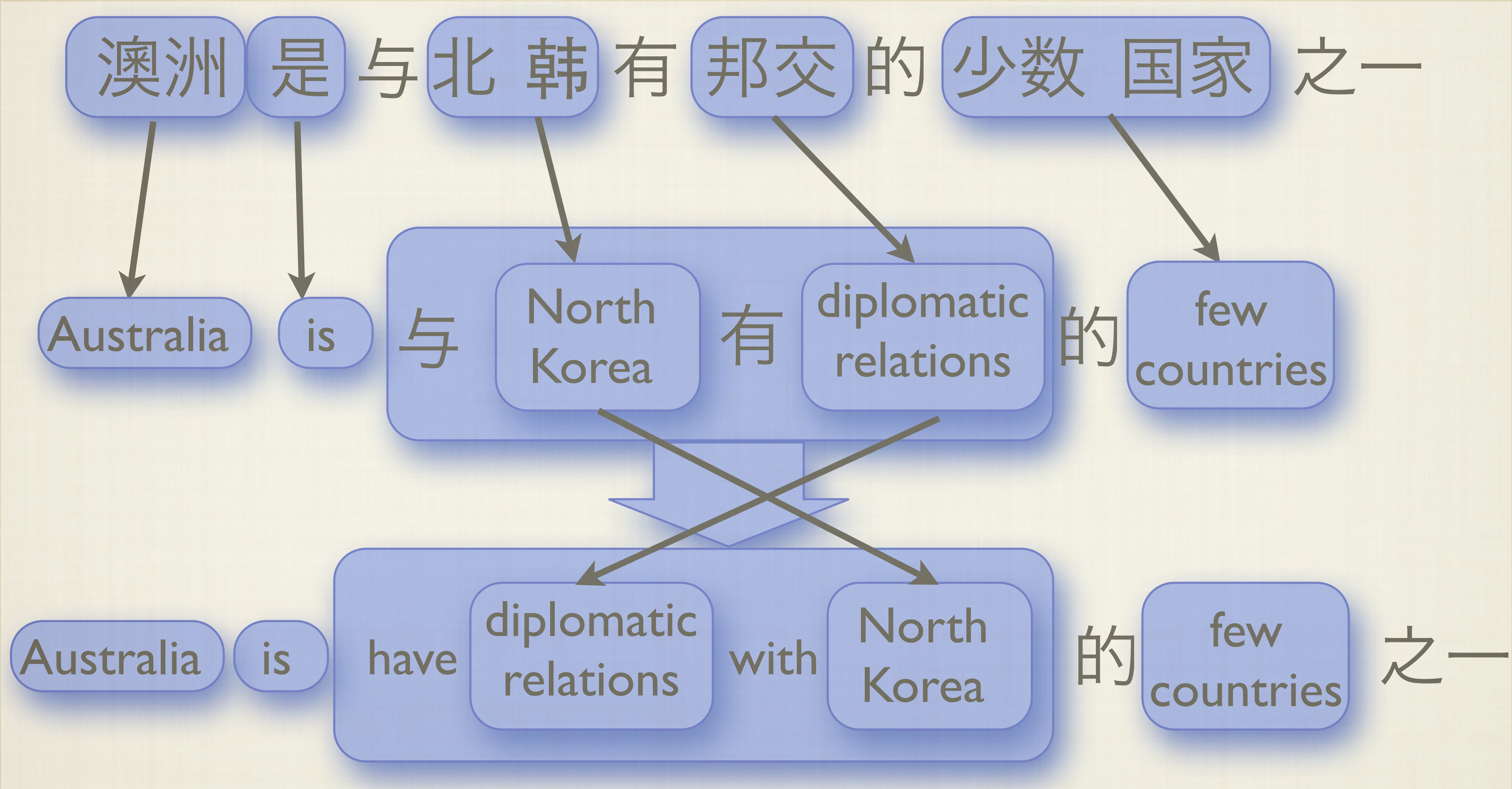
Hierarchical Translation



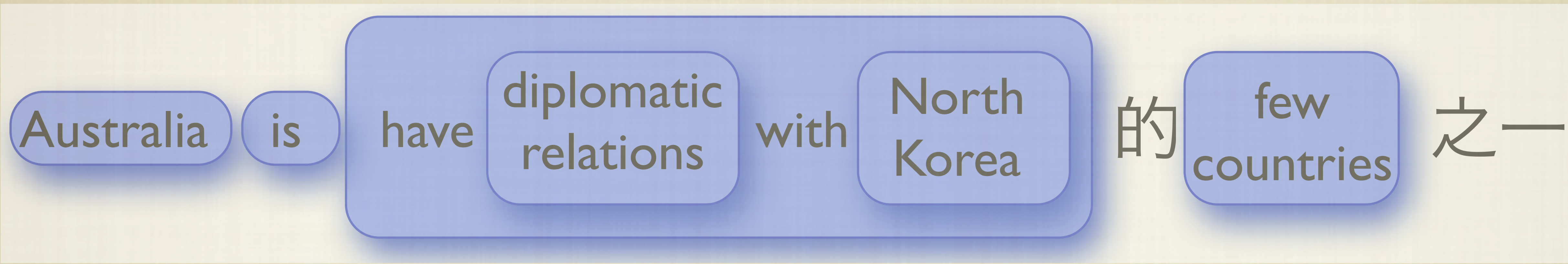
Hierarchical Translation



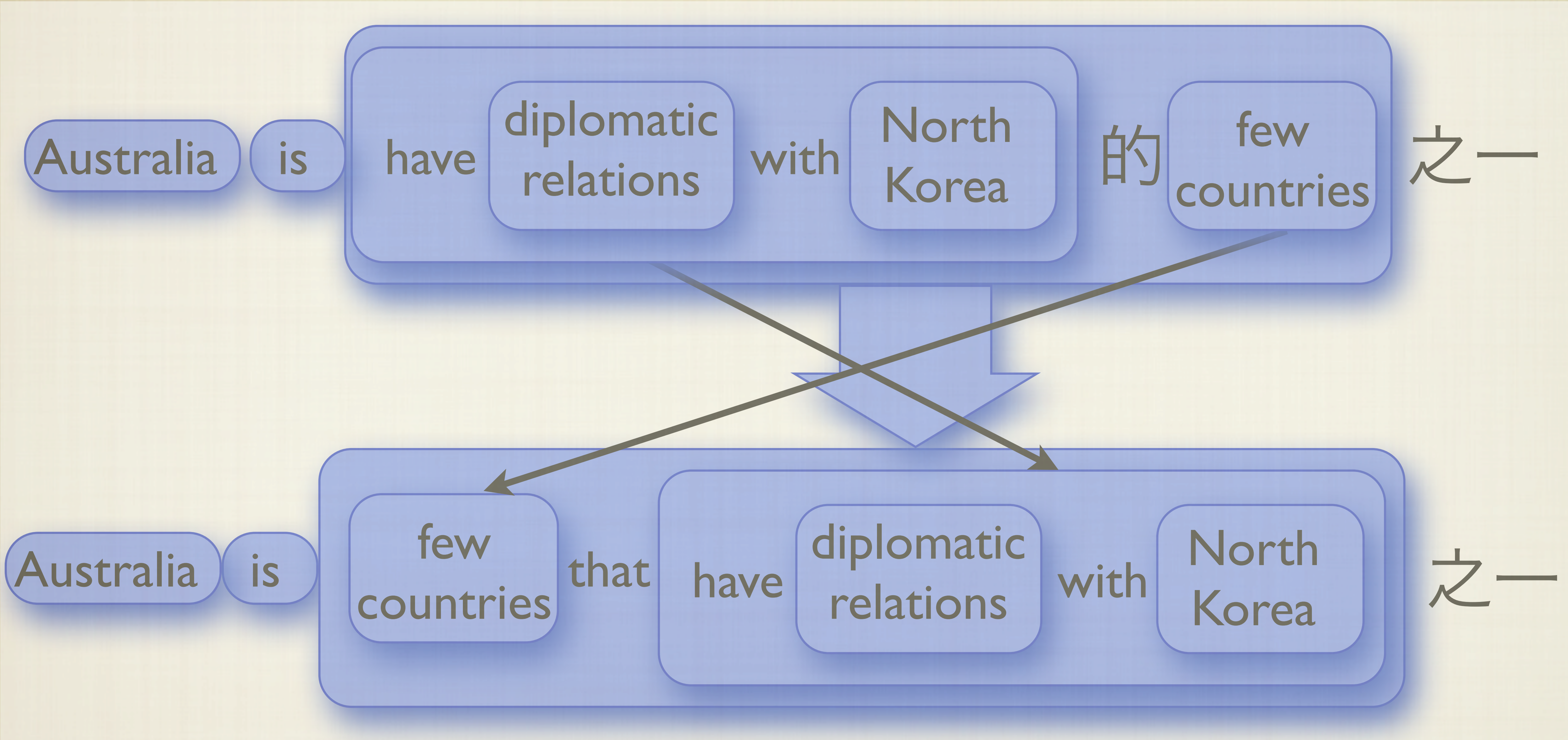
Hierarchical Translation



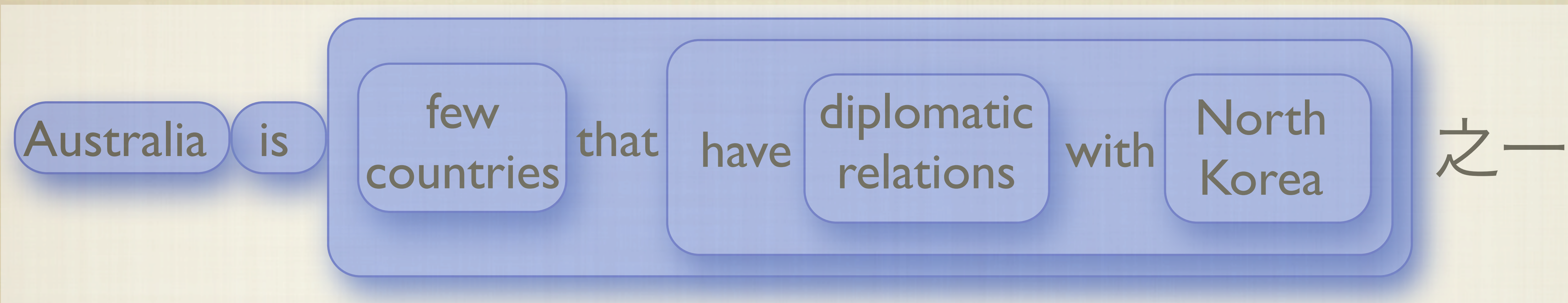
Hierarchical Translation



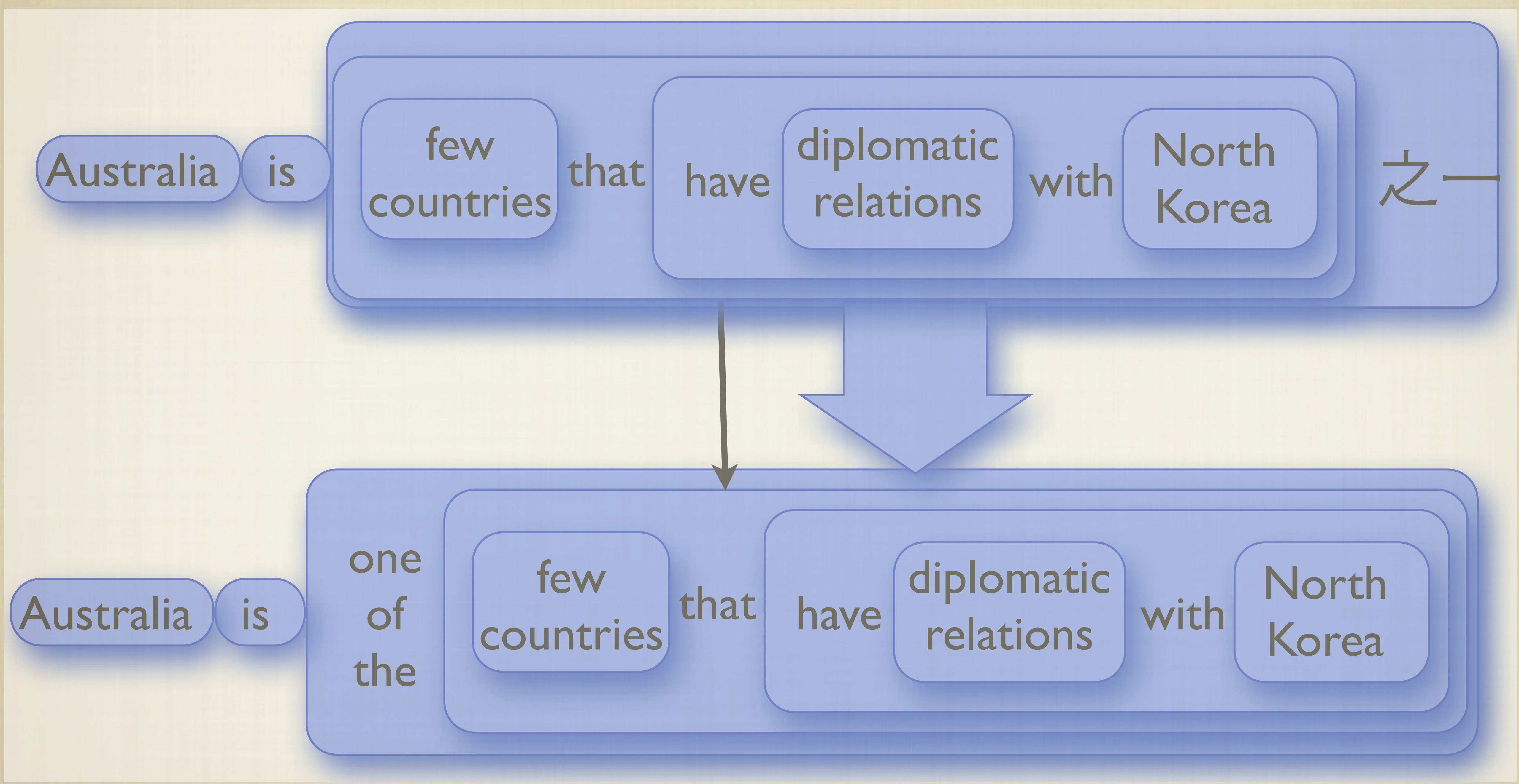
Hierarchical Translation



Hierarchical Translation



Hierarchical Translation



reordering problem we saw before is solved!

Hierarchical Rules

- **Extracted from phrases**

  (you know phrase extraction from previous lecture)

与中国有外交谈判 → have diplomatic talks with China

外交谈判 → diplomatic talks

中国 → China

Hierarchical Rules

- **Extracted from phrases**

  (you know phrase extraction from previous lecture)

与中国有外交谈判 → have diplomatic talks with China

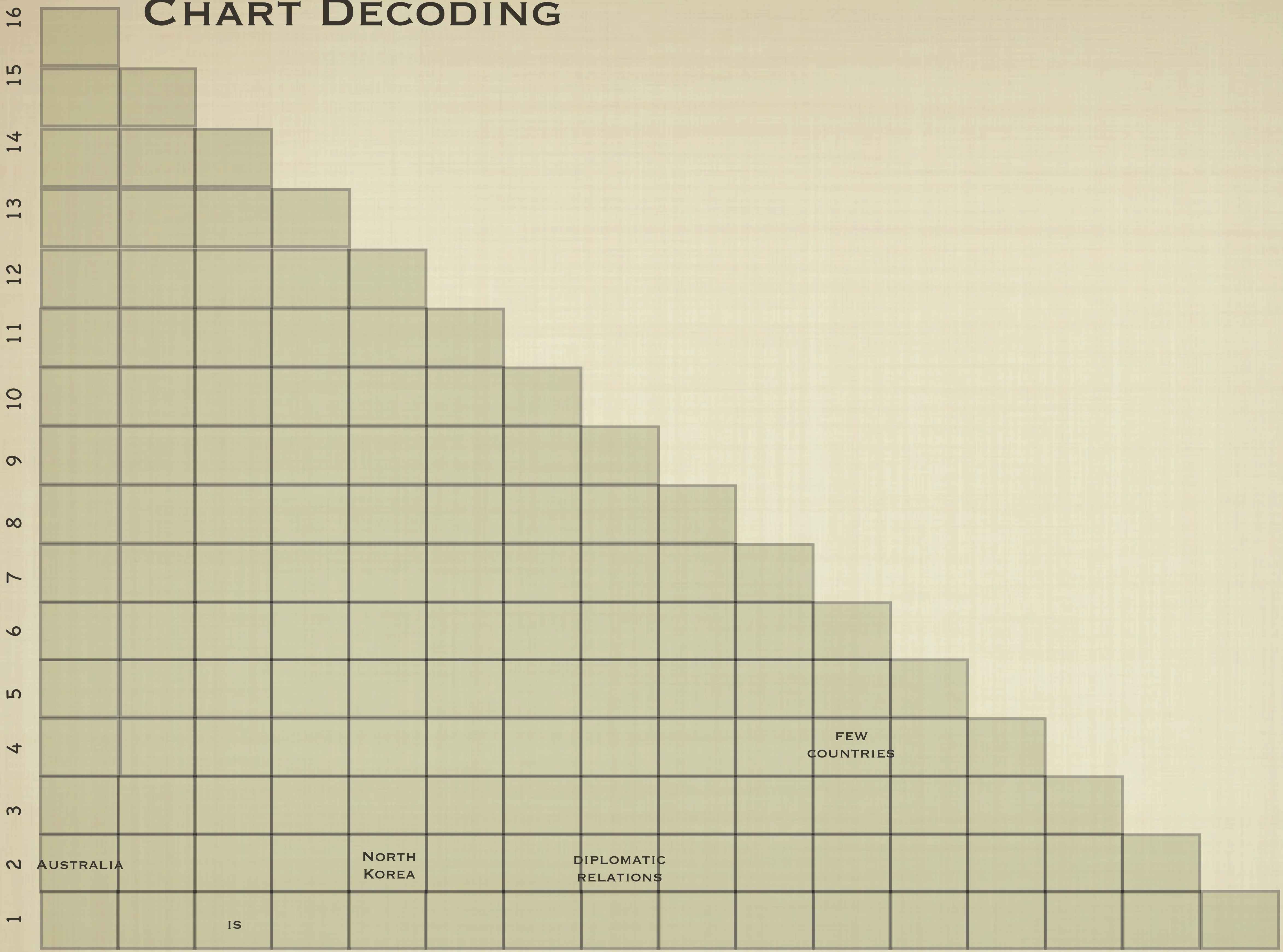
外交谈判 → diplomatic talks

中国 → China

↓

X: 与 [X1] 有 [X2] , have [X2] with [X1]

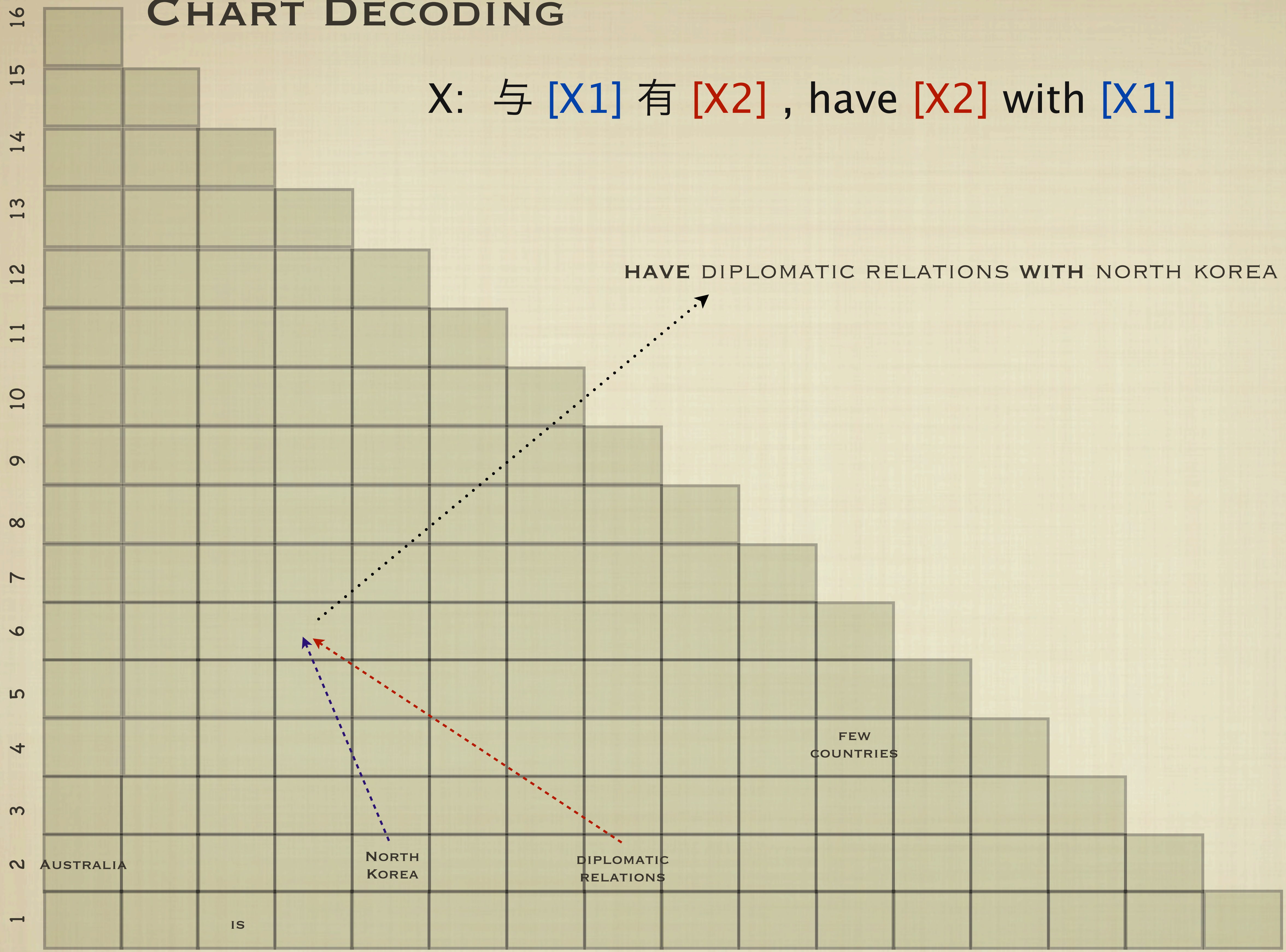
Chart Decoding



澳 洲是与北韩有 邦交的少 数国家之一

Chart Decoding

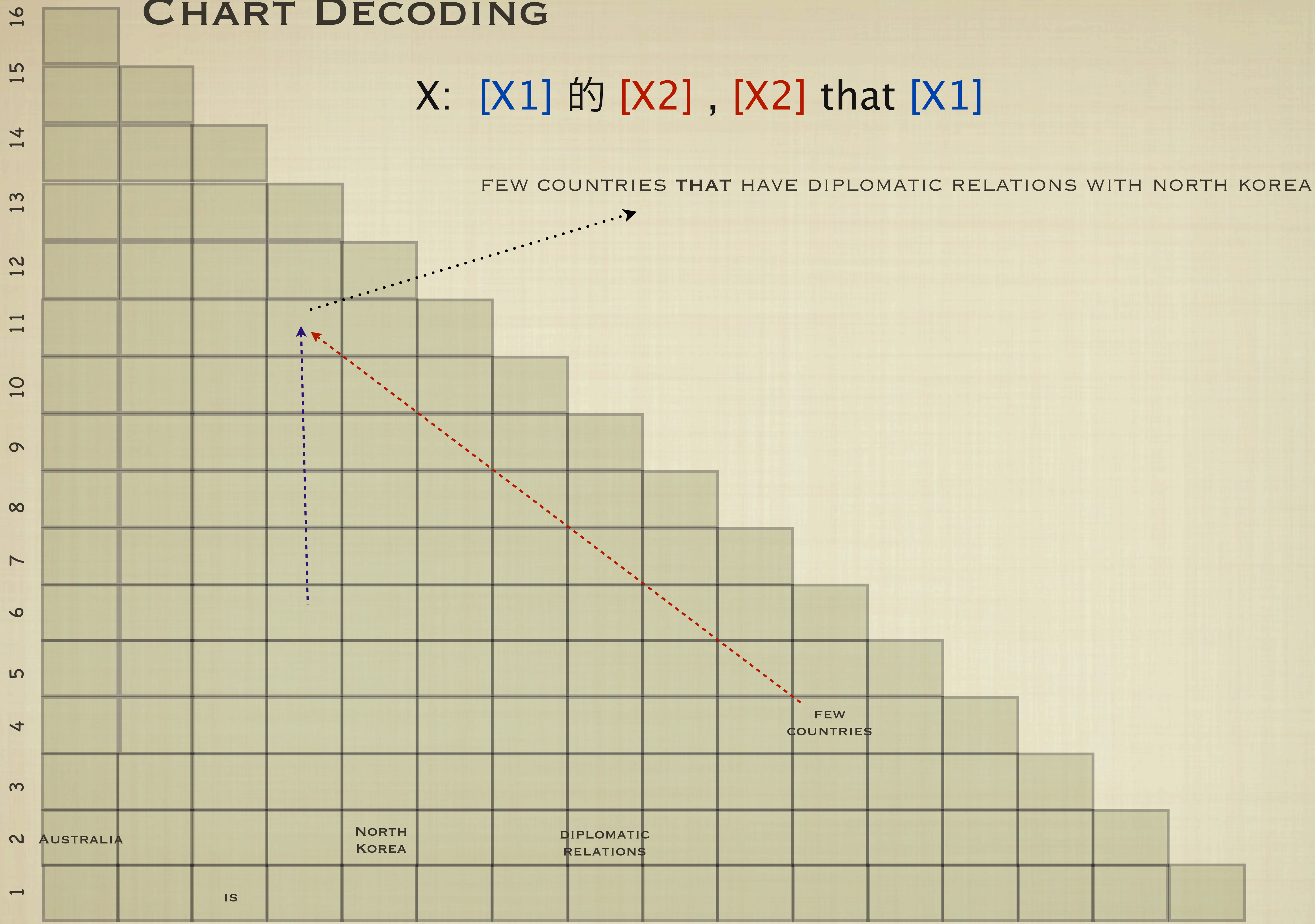
X: 与 [X1] 有 [X2] , have [X2] with [X1]



澳 洲是与北韩有 邦交的少 数国家之一

Chart Decoding

X: [X1] 的 [X2] , [X2] that [X1]



澳 洲是与北韩有 邦交的少 数国家之一

Chart Decoding

X: [X1] 之一 , one of the [X1]

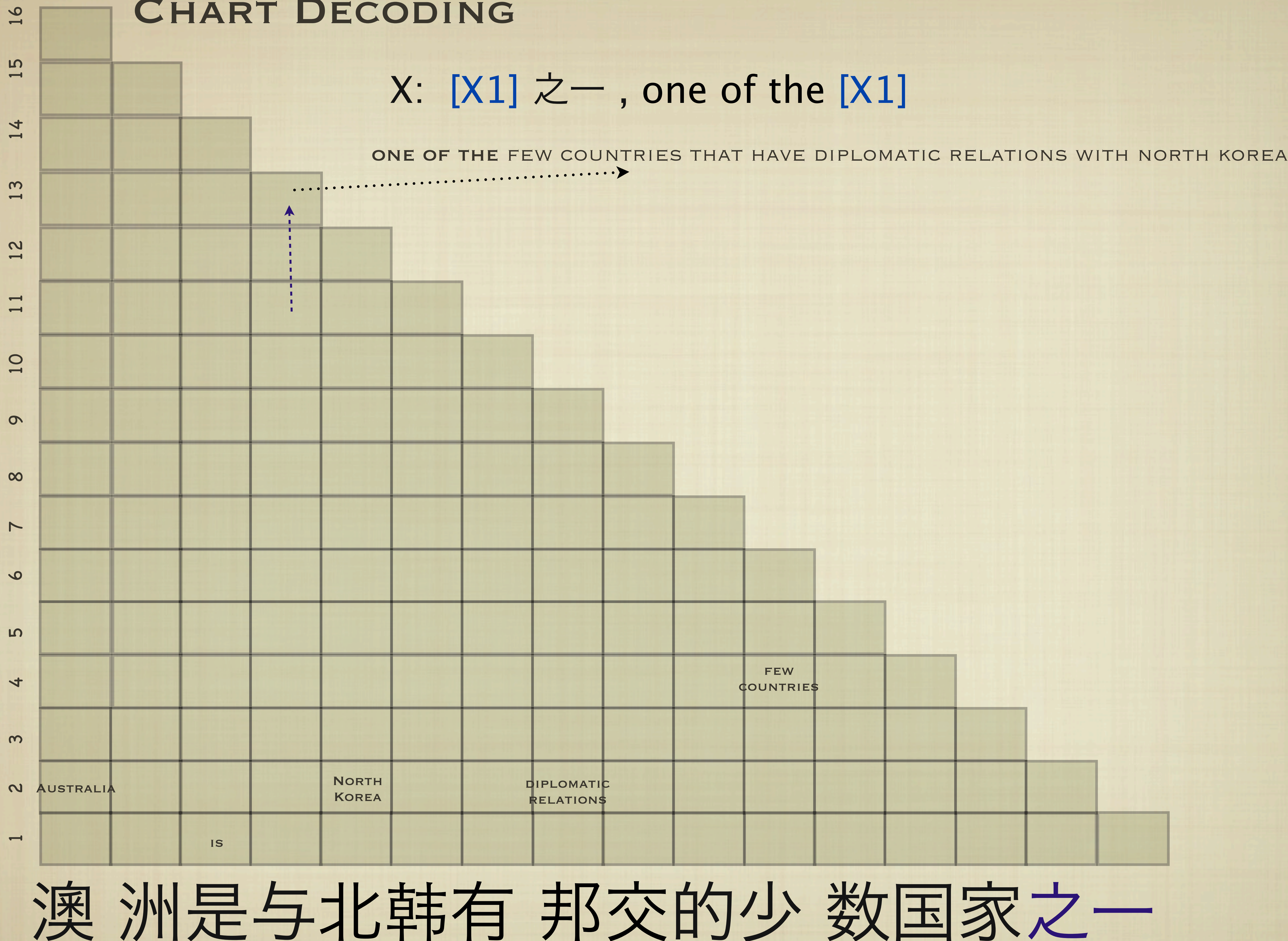


Chart Decoding

X: [X1] [X2] , [X1] [X2]   "glue rule"

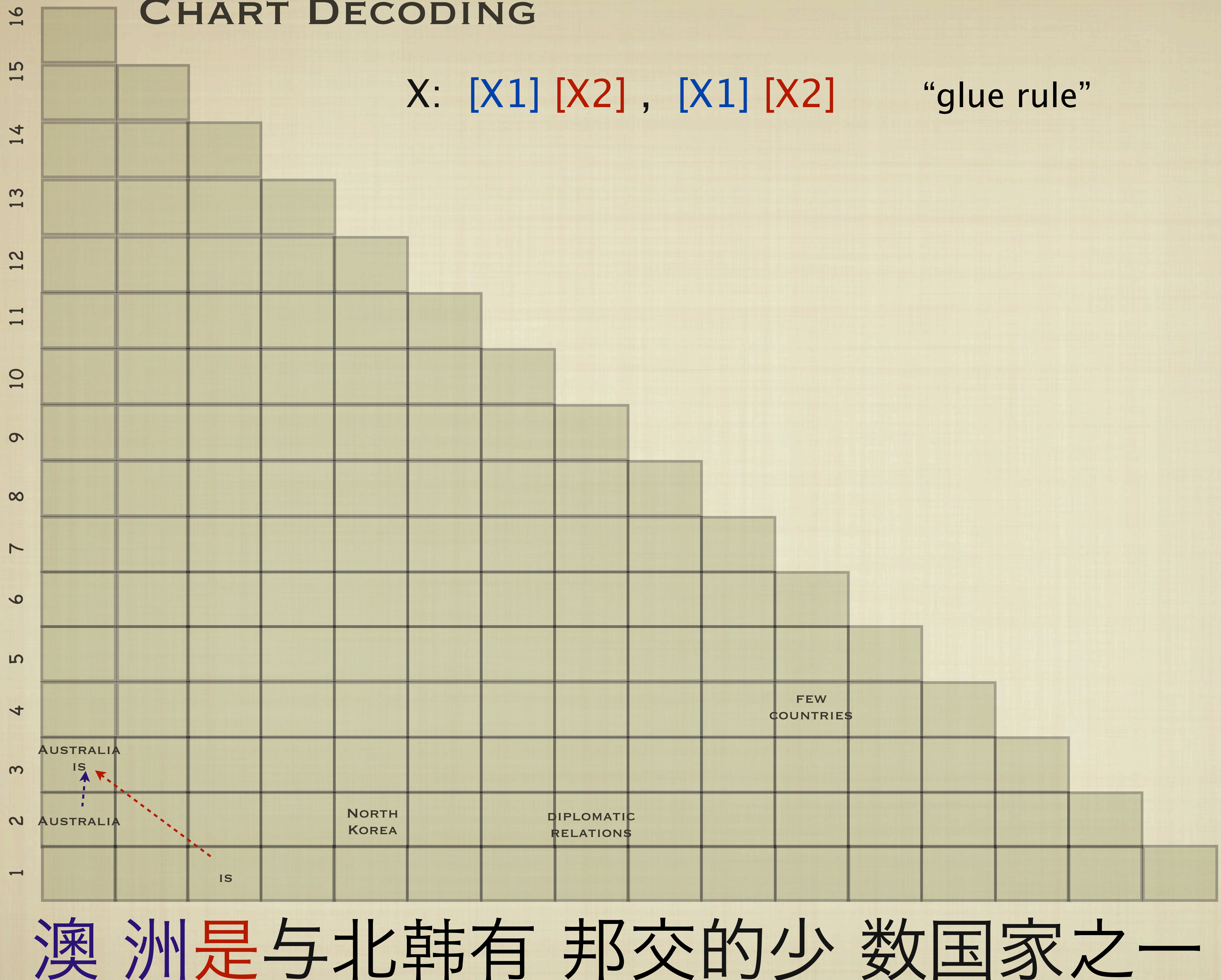
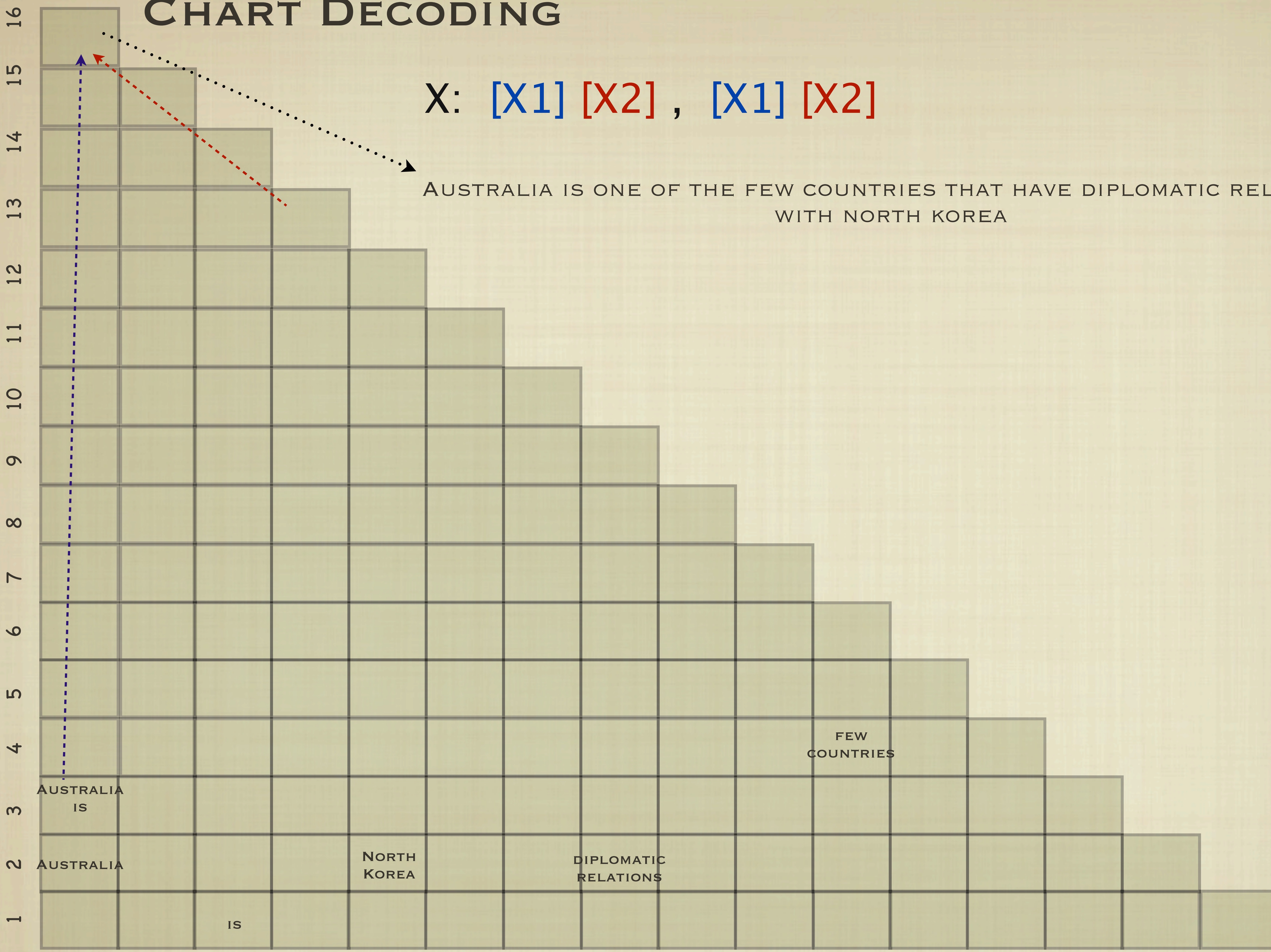


Chart Decoding

X: [X1] [X2] , [X1] [X2]

Australia is one of the few countries that have diplomatic relations with North Korea



澳 洲是与北韩有 邦交的少 数国家之一

Chart Decoding: Complexity

$$n\frac{(n+1)}{2}h^2r$$

cells

new hyps per each rule in each cell

n : sentence length (in 10s)

h : # kept hyps in each cell (in 100s)

r : # rules in each cell (in 10s)

→ h^2 contributes the most to the complexity

Chart Decoding: Cube Pruning

Problem: Find the best 5 combined hyps without trying all 25 possible combinations

	h1	h2	h3	h4	h5
g1					
g2					
g3					
g4					
g5					

X: 与 [X1] 有 [X2] , have [X2] with [X1]

Chart Decoding: Cube Pruning

- The hyps in the lower cell are ordered
- But: $h_1 < h_2, g_1 < g_2 \nRightarrow (h_1 + g_1) < (h_2 + g_2)$
- We need a strategy to explore the most promising combinations

Chart Decoding: Cube Pruning

- Why $h_1 < h_2, g_1 < g_2 \nRightarrow (h_1 + g_1) < (h_2 + g_2)$?

- Main reason is language model (or any context-sensitive model)

- P(is) < P(was) doesn't necessarily mean P(Australia is) < P(Australia was)

P(Australia is) = P(Australia)+P(is| Australia)

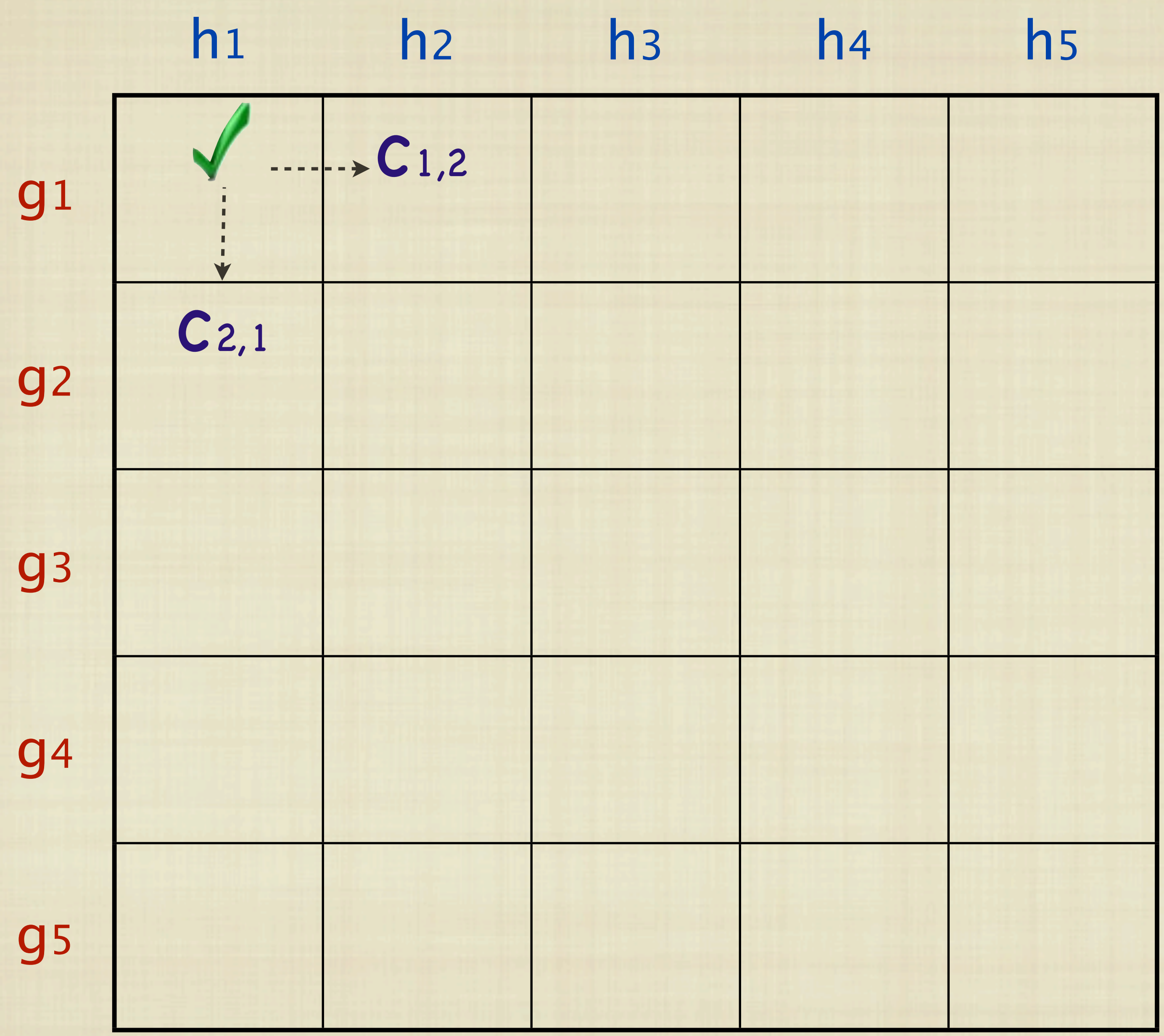
X: [X1] [X2] , [X1] [X2]

P(Australia was) = P(Australia)+P(was| Australia)

- Context changes. True for all language models orders except unigram

Chart Decoding: Cube Pruning

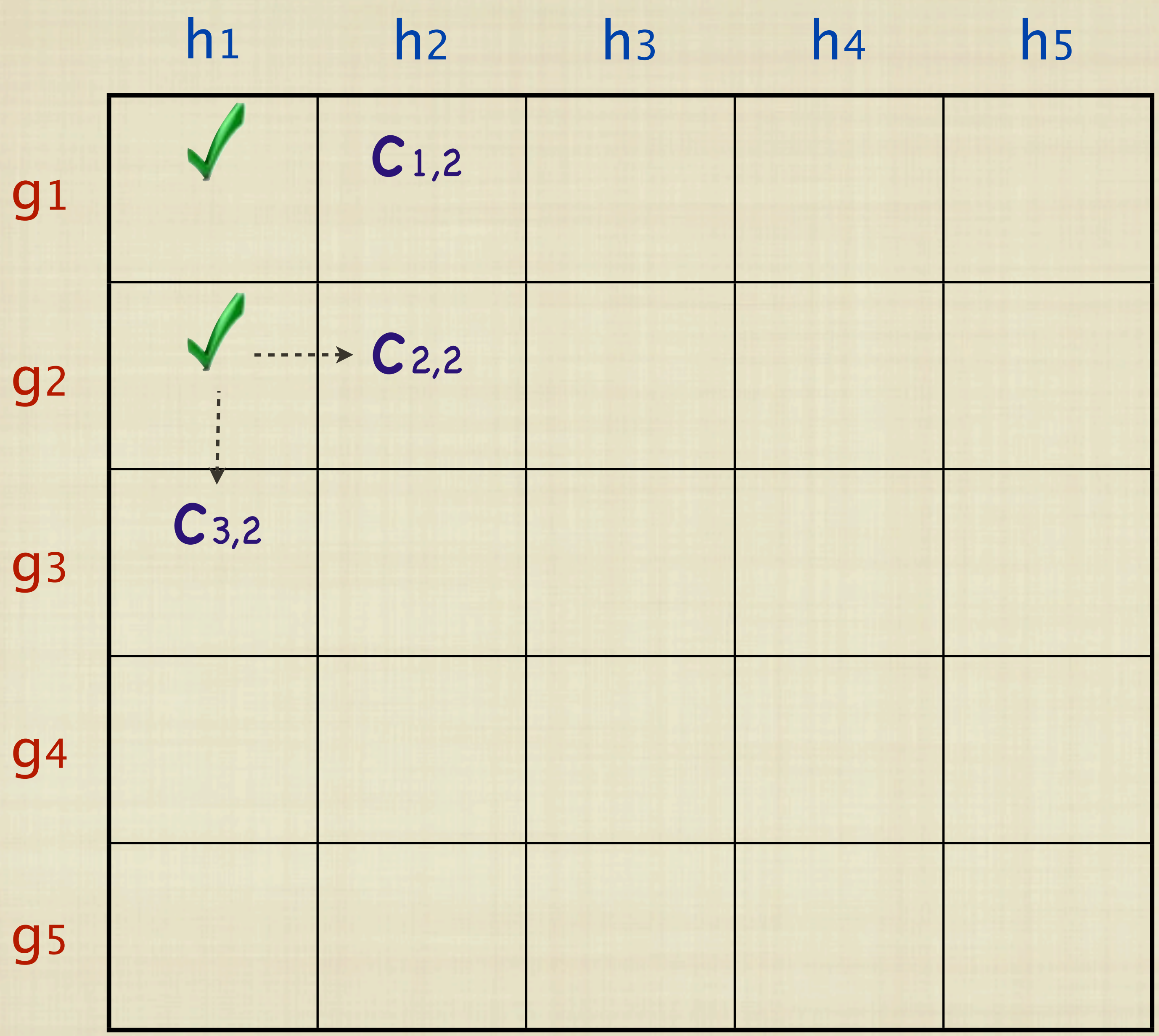
- Start with the top-right corner, build the new hyp, and add the two neighboring combinations to the priority list

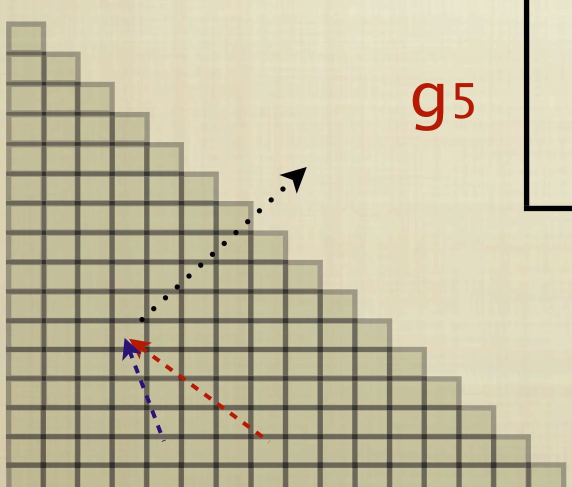


X: 与 [X1] 有 [X2] , have [X2] with [X1]

Chart Decoding: Cube Pruning

- Pick the lowest cost item in the priority queue, build hyp and add neighbors
- Priority queue ordered by approximate cost: $c_{x,y} = h_x + g_y$

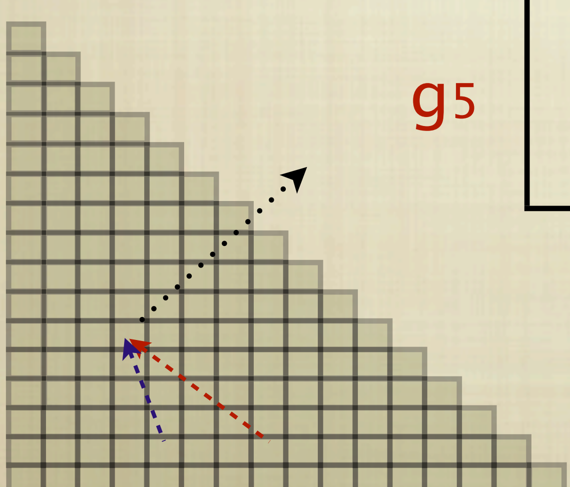




X: 与 [X1] 有 [X2] , have [X2] with [X1]

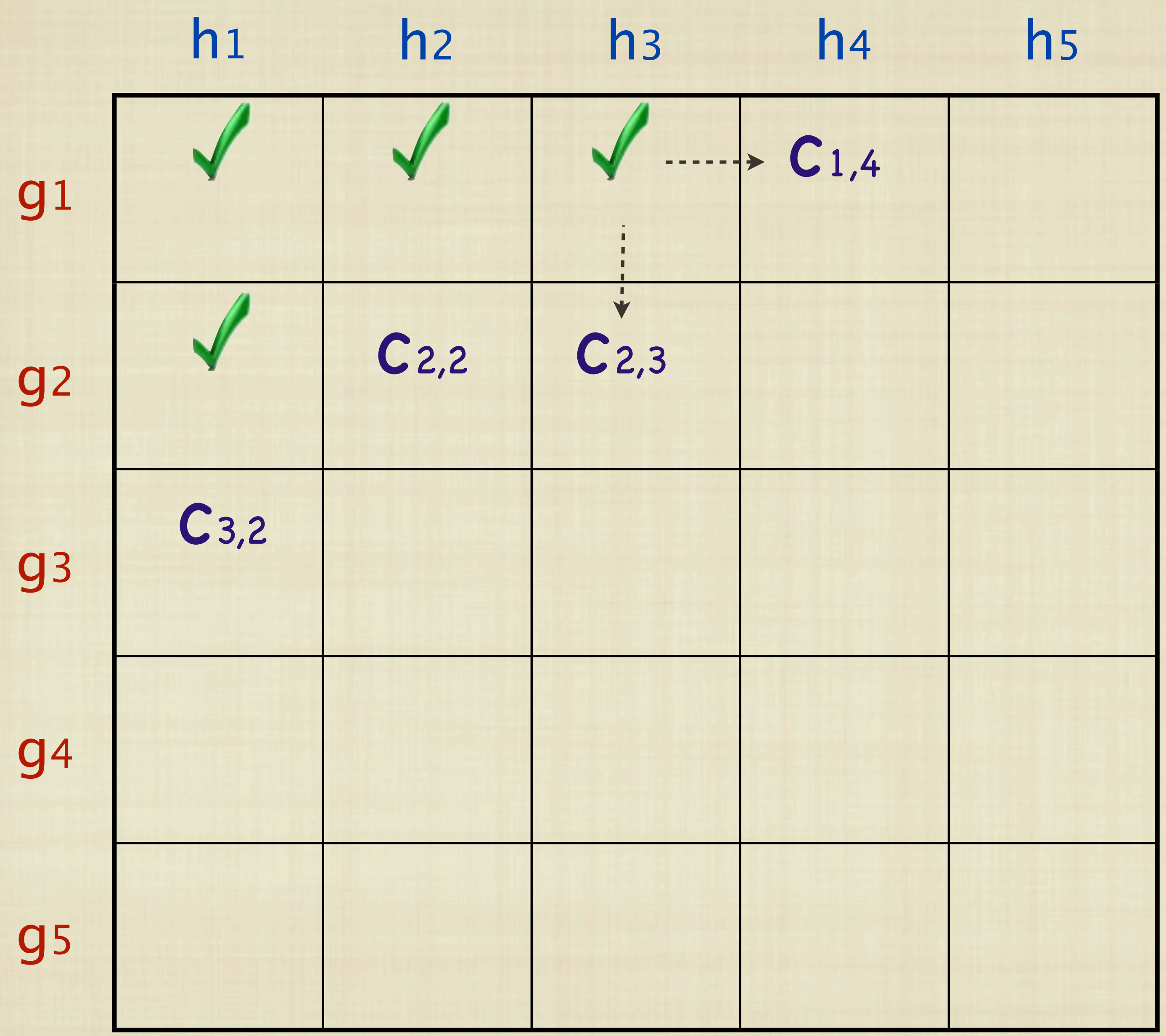
Chart Decoding: Cube Pruning

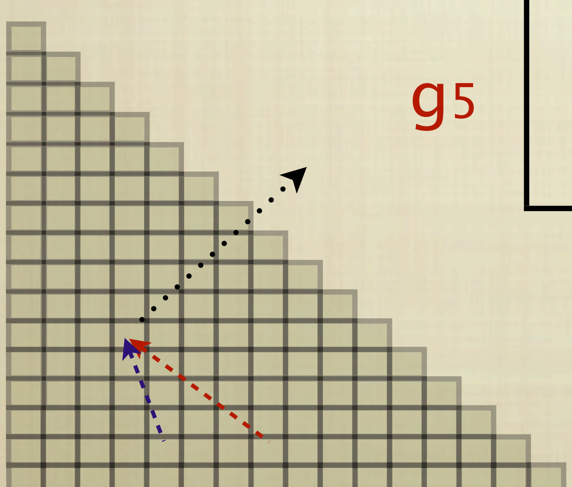
	h1	h2	h3	h4	h5
g1	✓	✓ ⇢	$c_{1,3}$		
g2	✓	$c_{2,2}$			
g3	$c_{3,2}$				
g4					
g5					



X: 与 [X1] 有 [X2] , have [X2] with [X1]

Chart Decoding: Cube Pruning

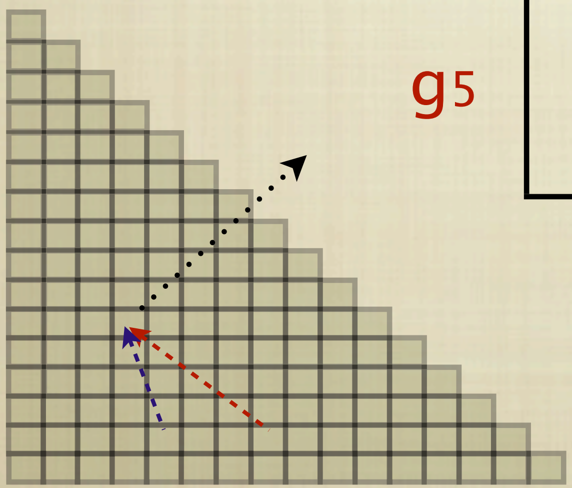




X: 与 [X1] 有 [X2] , have [X2] with [X1]

Chart Decoding: Cube Pruning

	h1	h2	h3	h4	h5
g1	✓	✓	✓	$C_{1,4}$	
g2	✓	$C_{2,2}$	✓ →	$C_{2,4}$	
g3	$C_{3,2}$		$C_{3,3}$		
g4					
g5					



X: 与 [X1] 有 [X2] , have [X2] with [X1]

Chart Decoding: Cube Pruning

Stopping criteria:

- Stop when a pre-defined number of hyps are generated (green checkmarks)
- Stop when the cost difference between the most recent hyp and the best one so far is higher than a pre-defined value

- Dynamic Programming
- Combine two hyps if their completion costs are the same
    - ie. however we extend one, we can extend the other in the same exact way, and with the same exact additional cost
- Define "states" based on the completion concept
- Combine hyps that share the same state

Chart Decoding: Hyp Combination

- Only compare and define hyps in the same cell (same coverage of foreign words)
- States are defined based on language model
  (or any additional context-sensitive models)
- We can append to a hyp from either left or right
- Language model state is defined as the (n-1) left-most and (n-1) right-most words

few countries that have diplomatic relations with north korea

Chart Decoding: Hyp Combination

- Cost of words in the middle are not going to change

  P(that) = P(that| few countries)

- Continuation costs depend on the last 2 words only

  P(. | North Korea)

  Does not depend on what words were there before "North Korea"

Chart Decoding: Hyp Combination

- Among the hyps in the same cell, sharing the same (LM) states, keep only the best (lowest-cost) one
- No need to explore the expansion of the higher-cost hyps with the same state
- The higher-cost hyps can be kept and used for building N-best lists

Search Space:

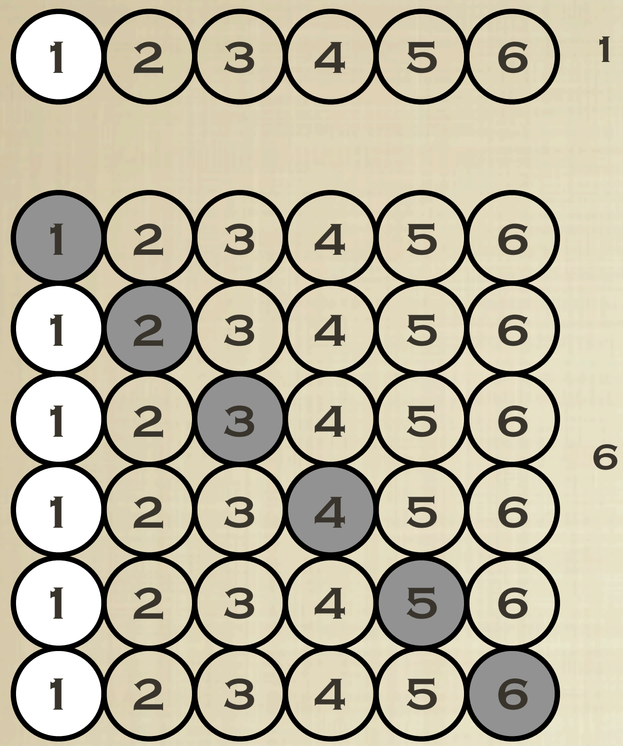
Phrase-bases vs. Hierarchical

Coverage Vectors: N=6, all possible permutations

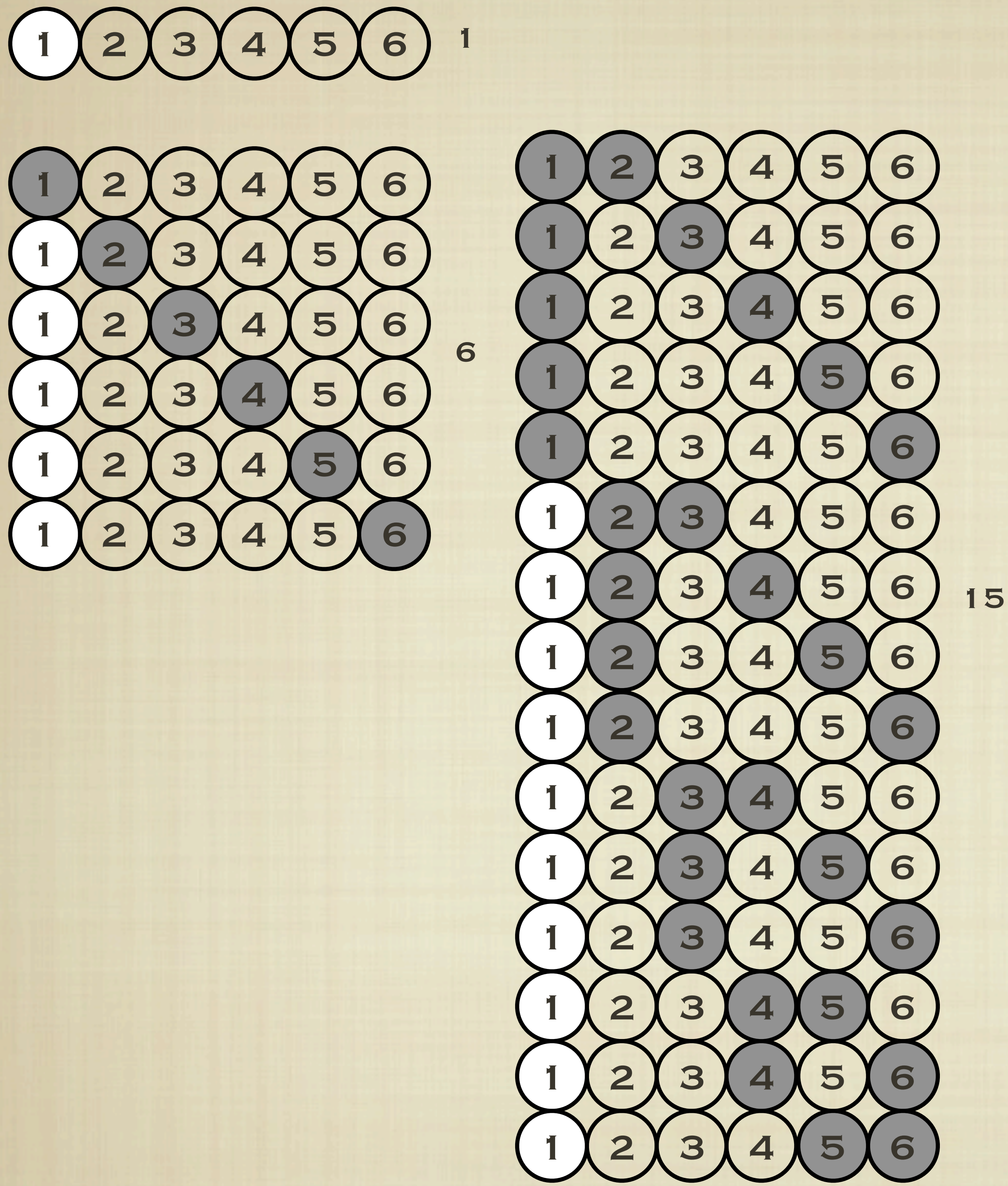
Coverage Vectors: N=6, all possible permutations

1

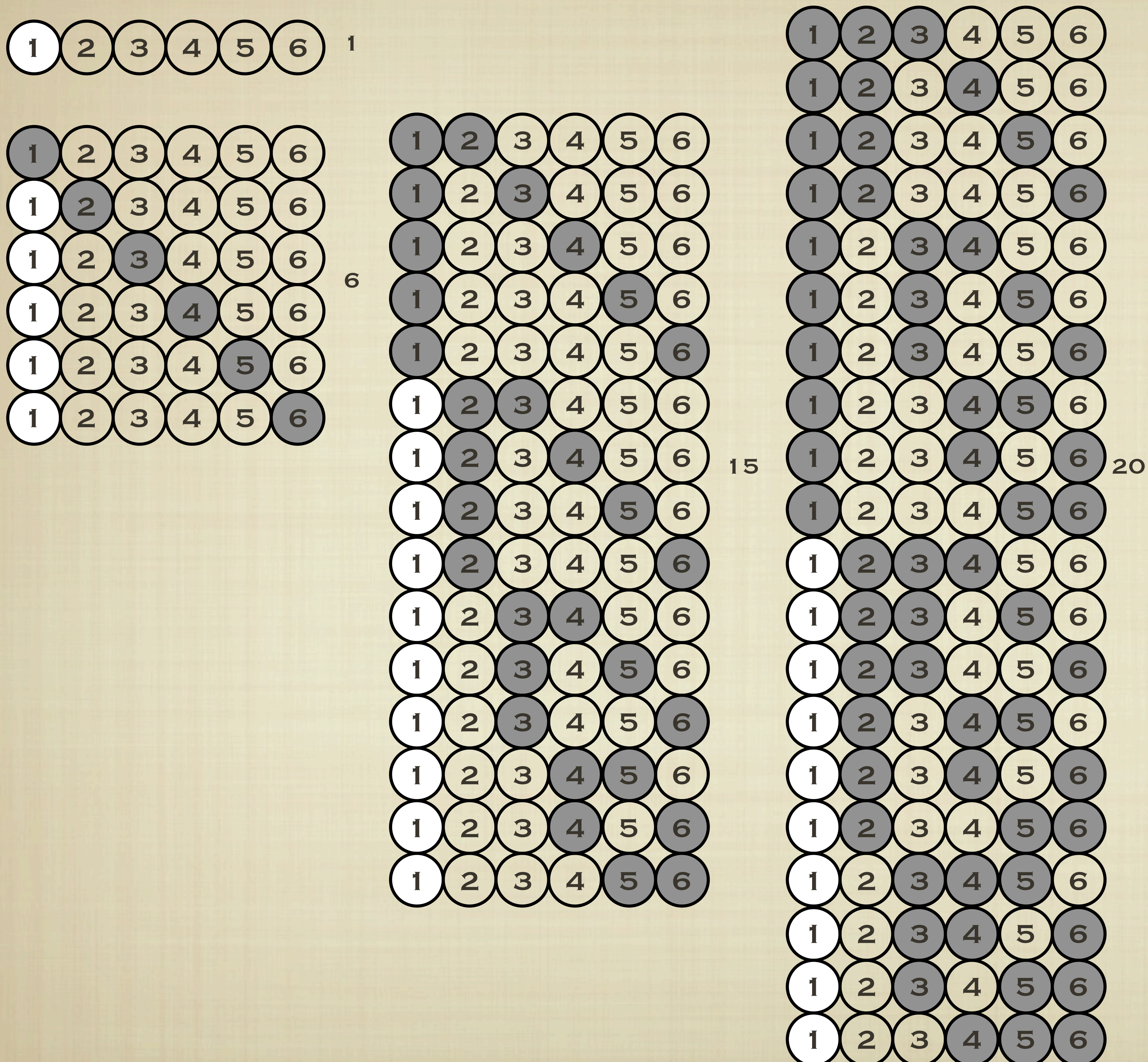
Coverage Vectors: N=6, all possible permutations



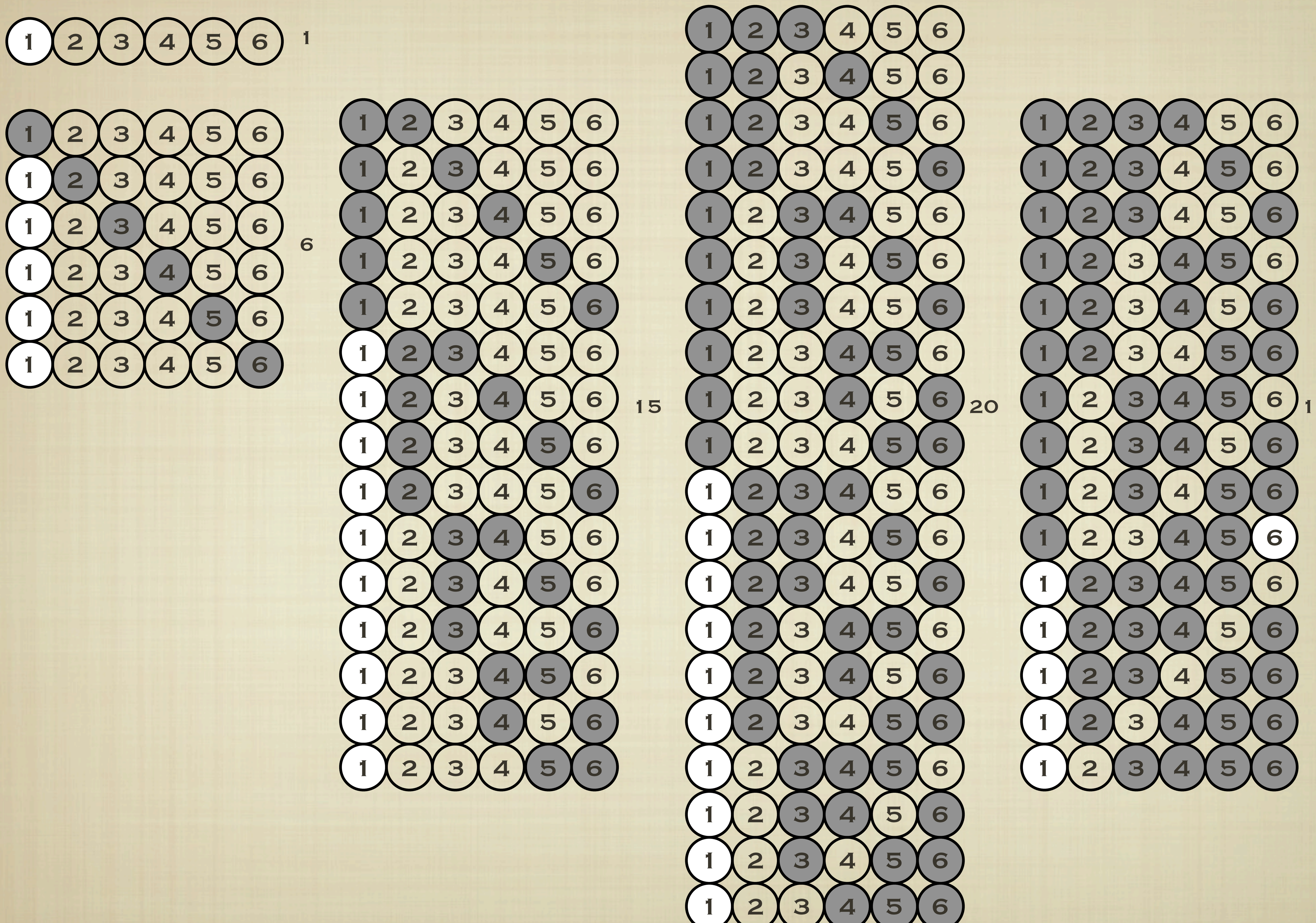
Coverage Vectors: N=6, all possible permutations



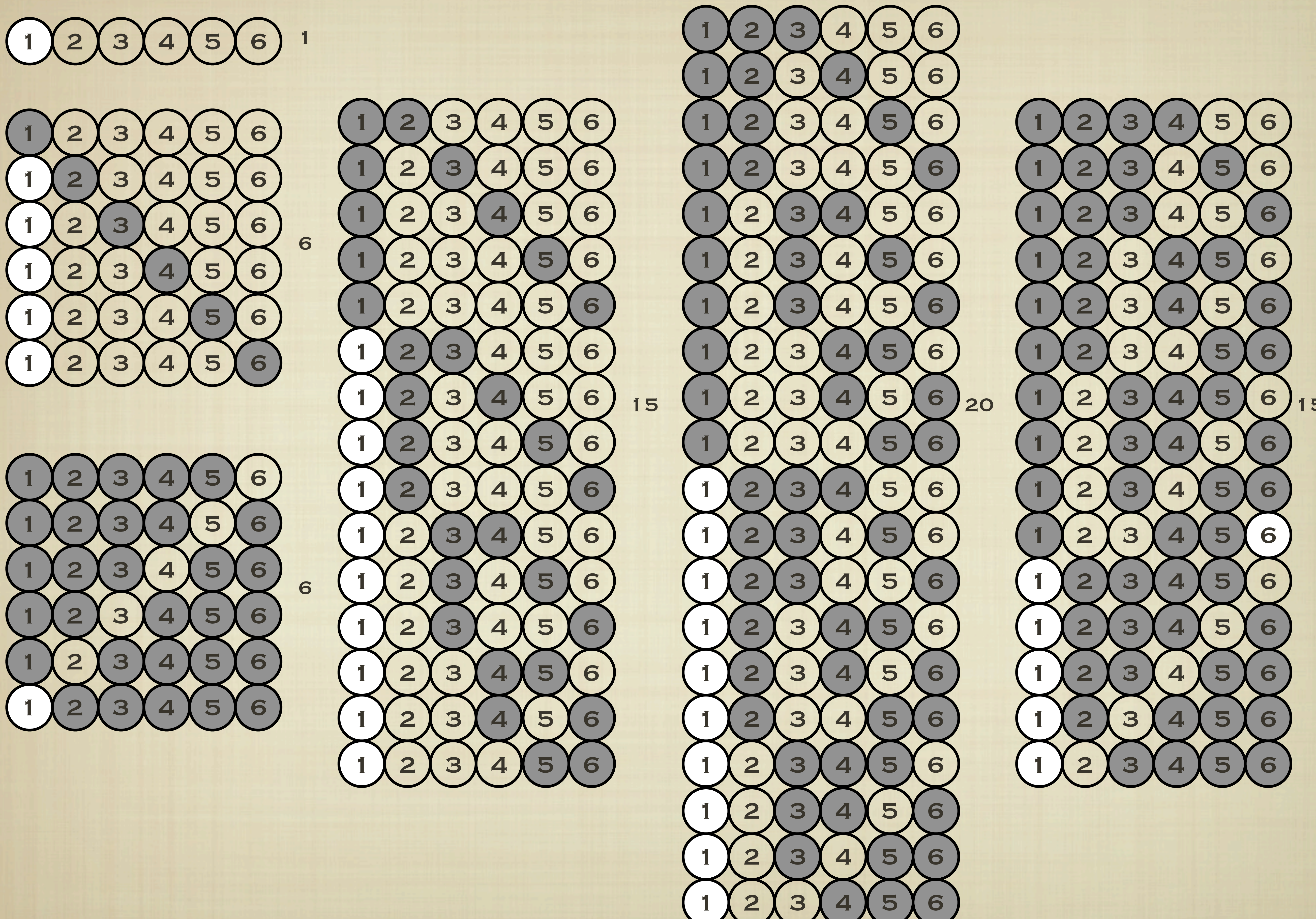
Coverage Vectors: N=6, all possible permutations



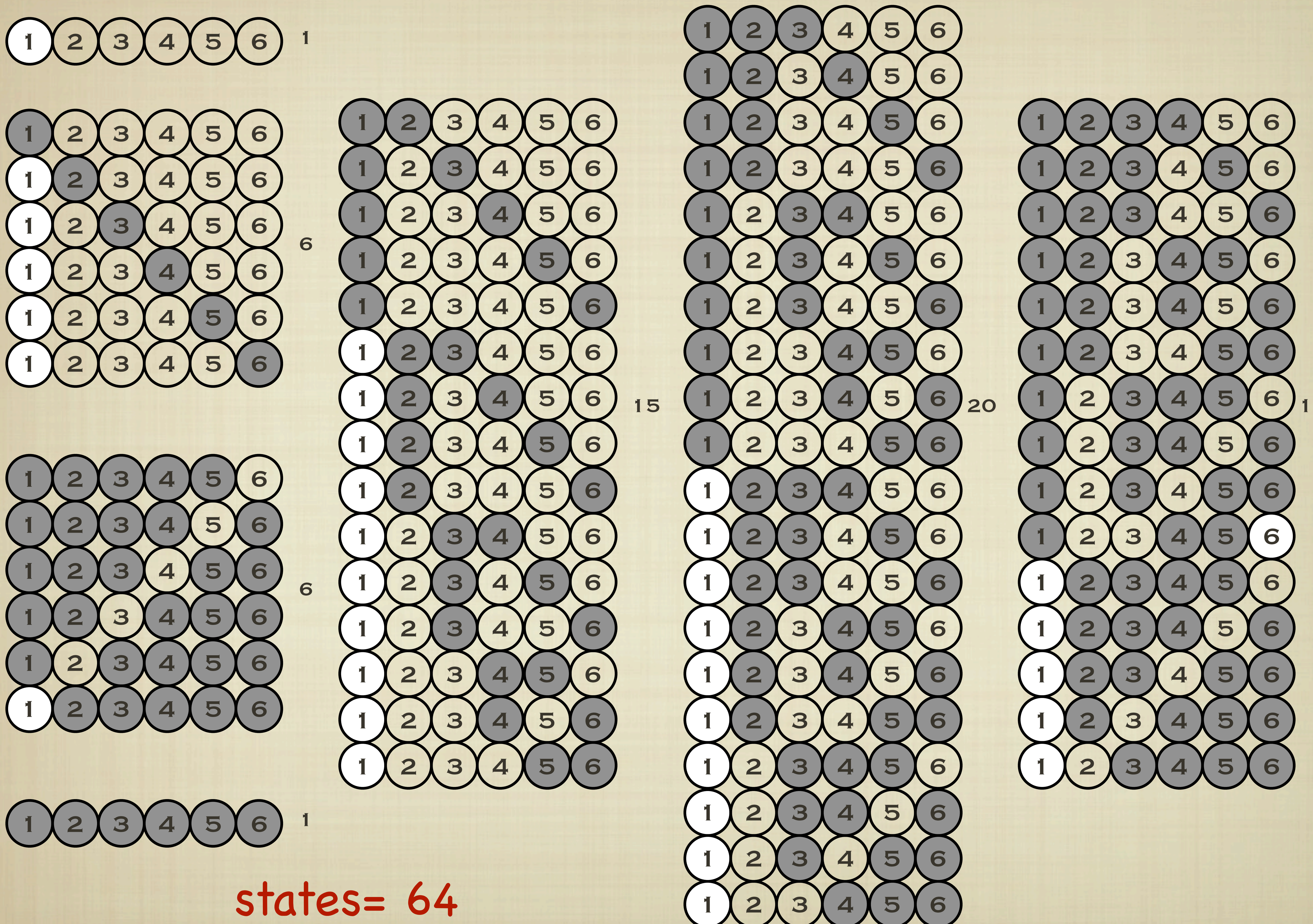
Coverage Vectors: N=6, all possible permutations



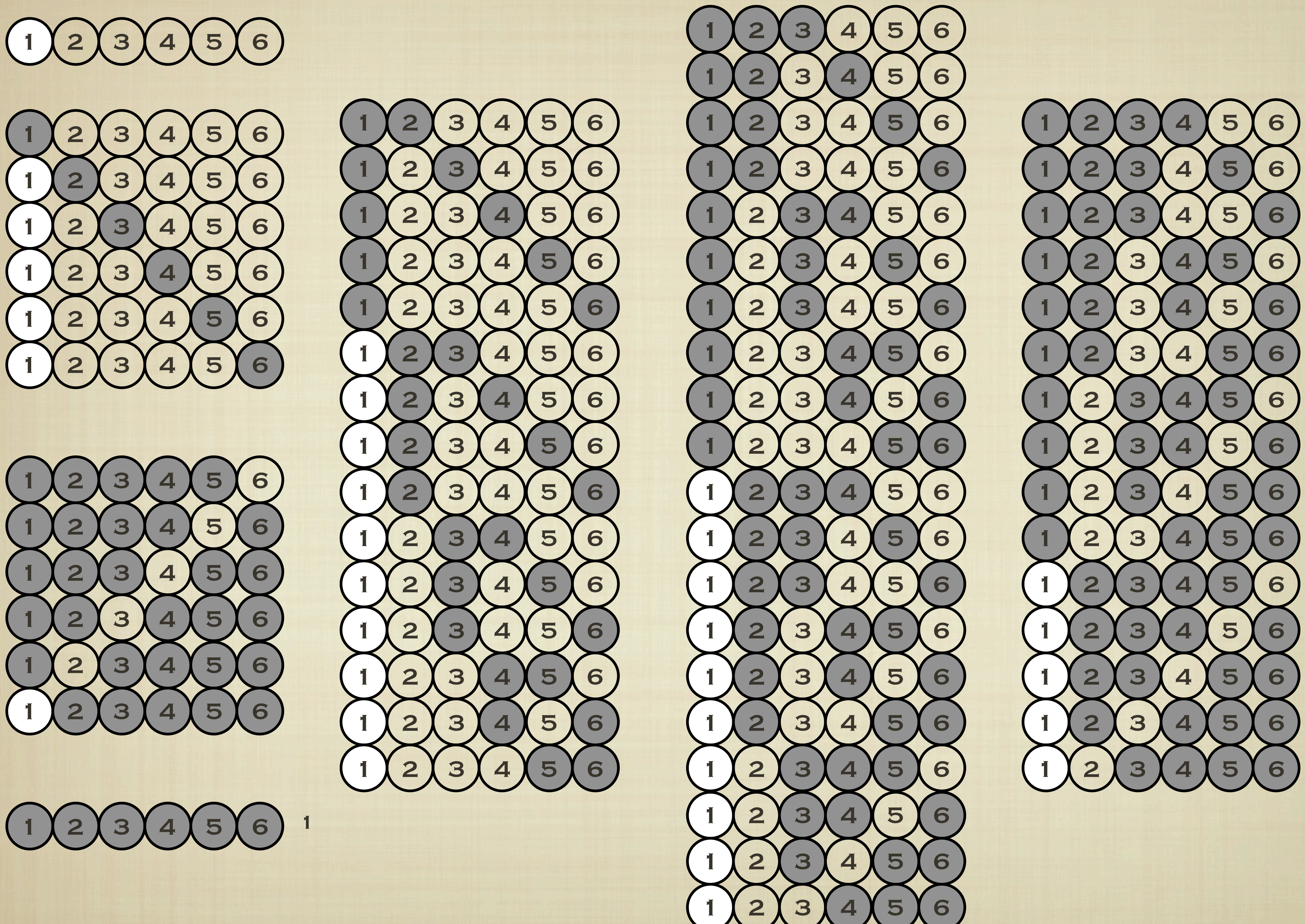
Coverage Vectors: N=6, all possible permutations



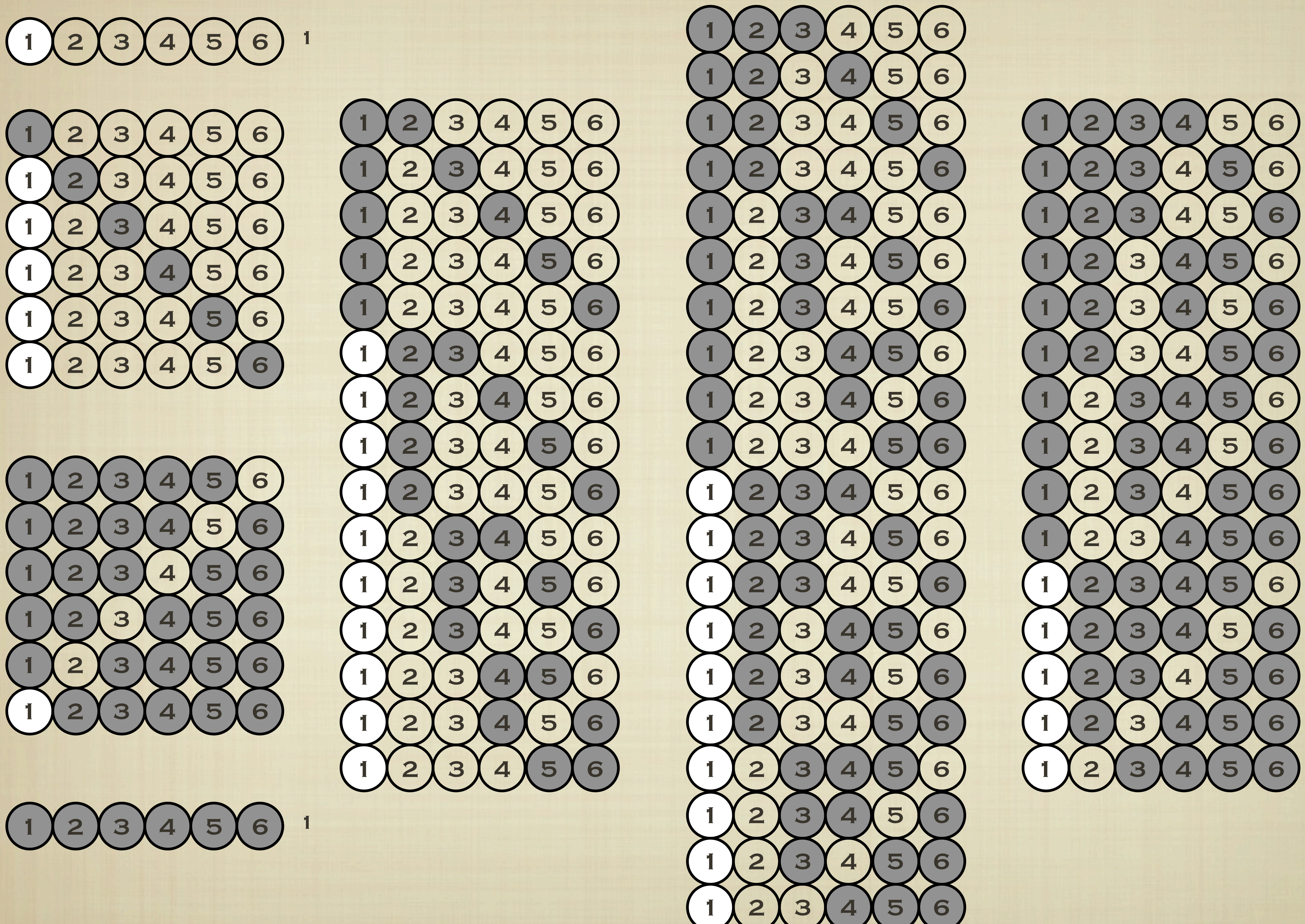
Coverage Vectors: N=6, all possible permutations



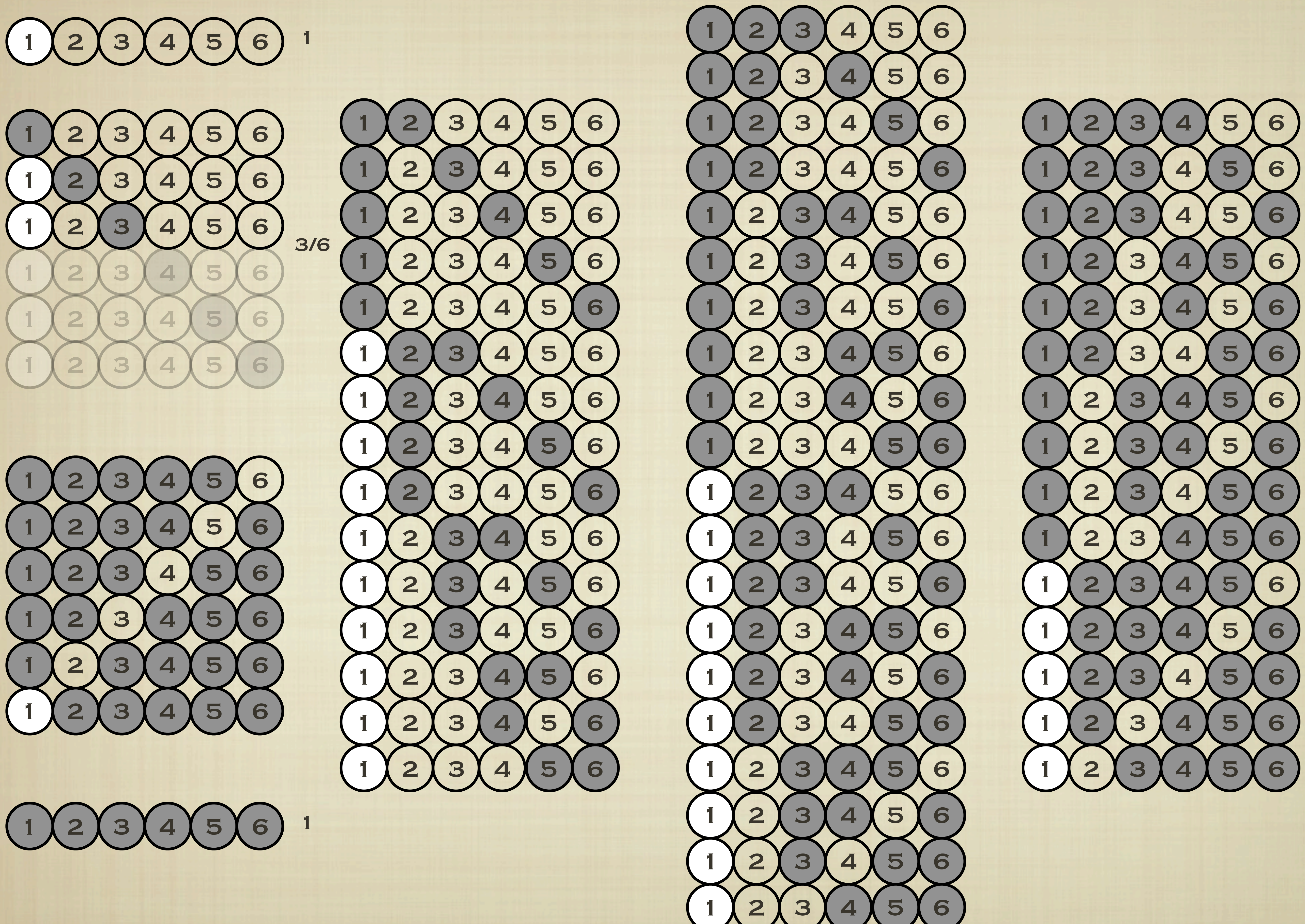
N=6, Phrase-based, S=2,WW=3



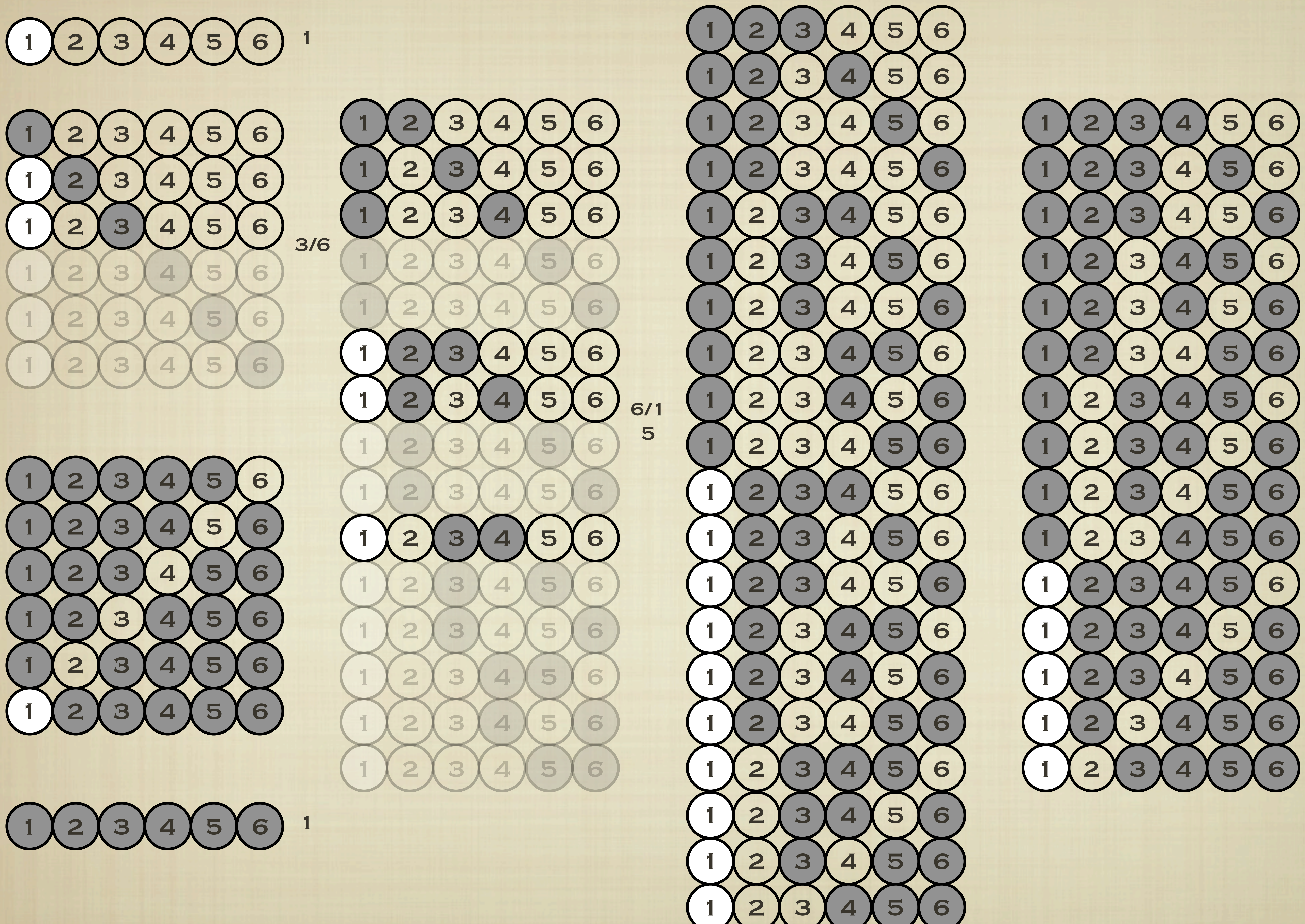
N=6, Phrase-based, S=2, WW=3



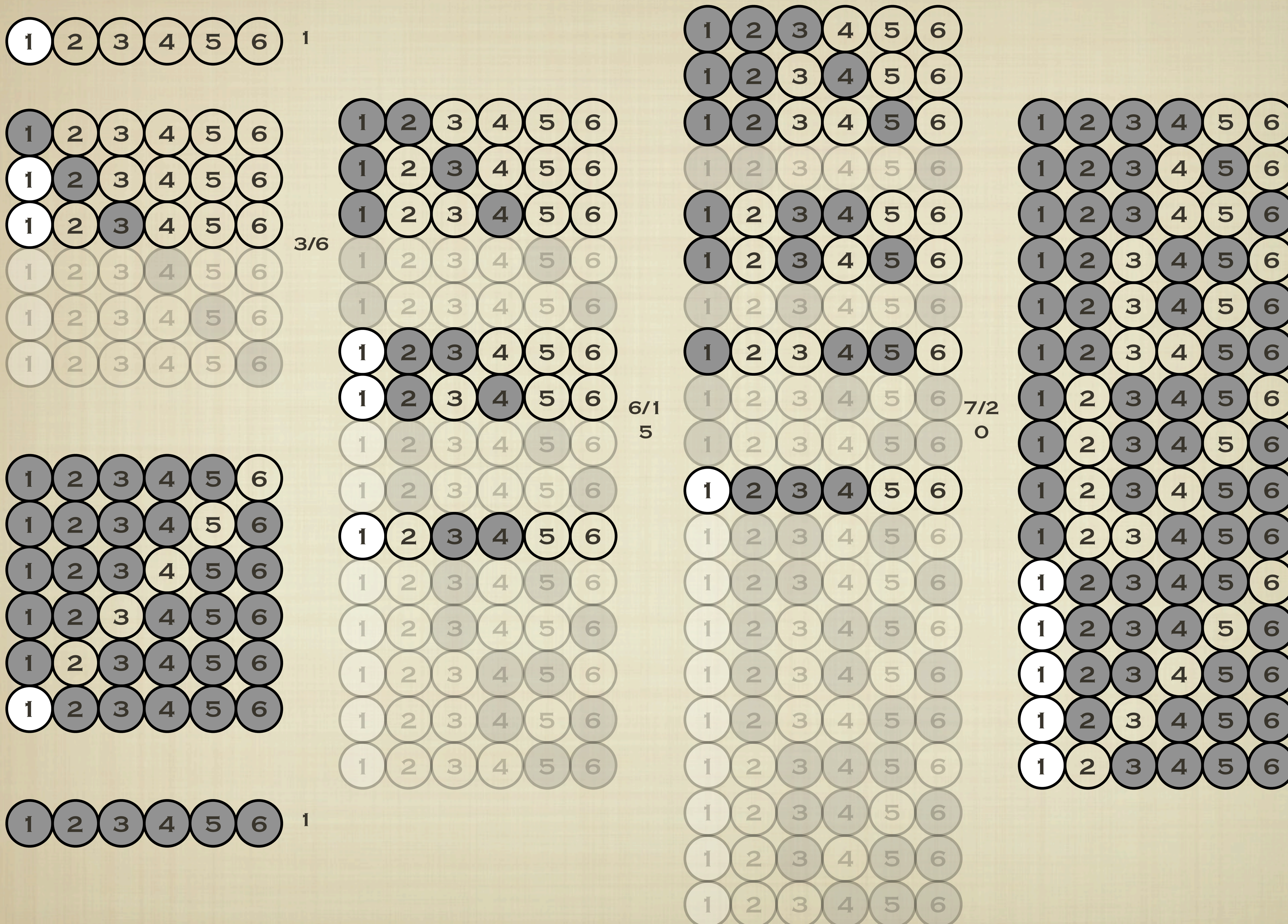
N=6, Phrase-based, S=2,WW=3



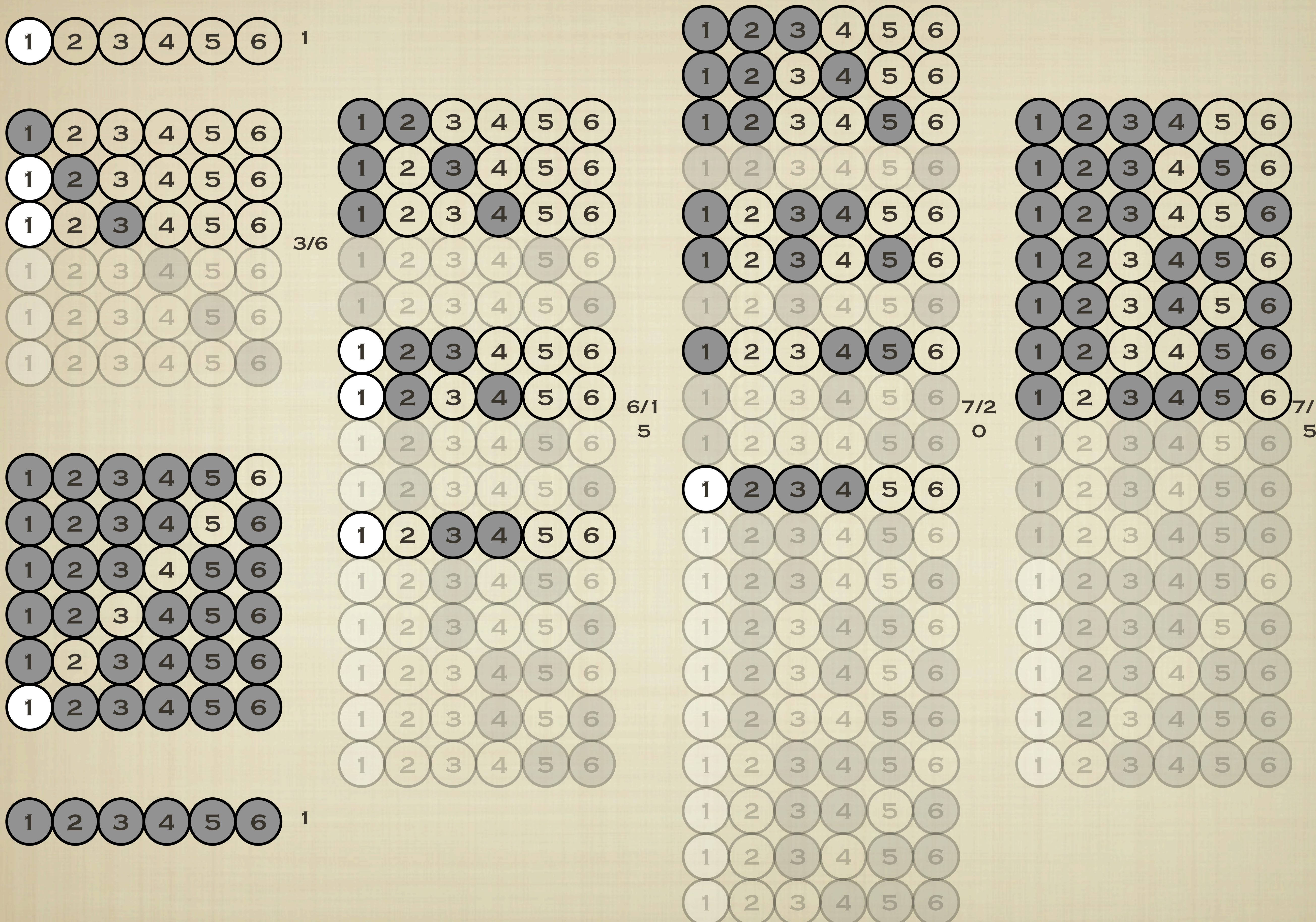
N=6, Phrase-based, S=2,WW=3



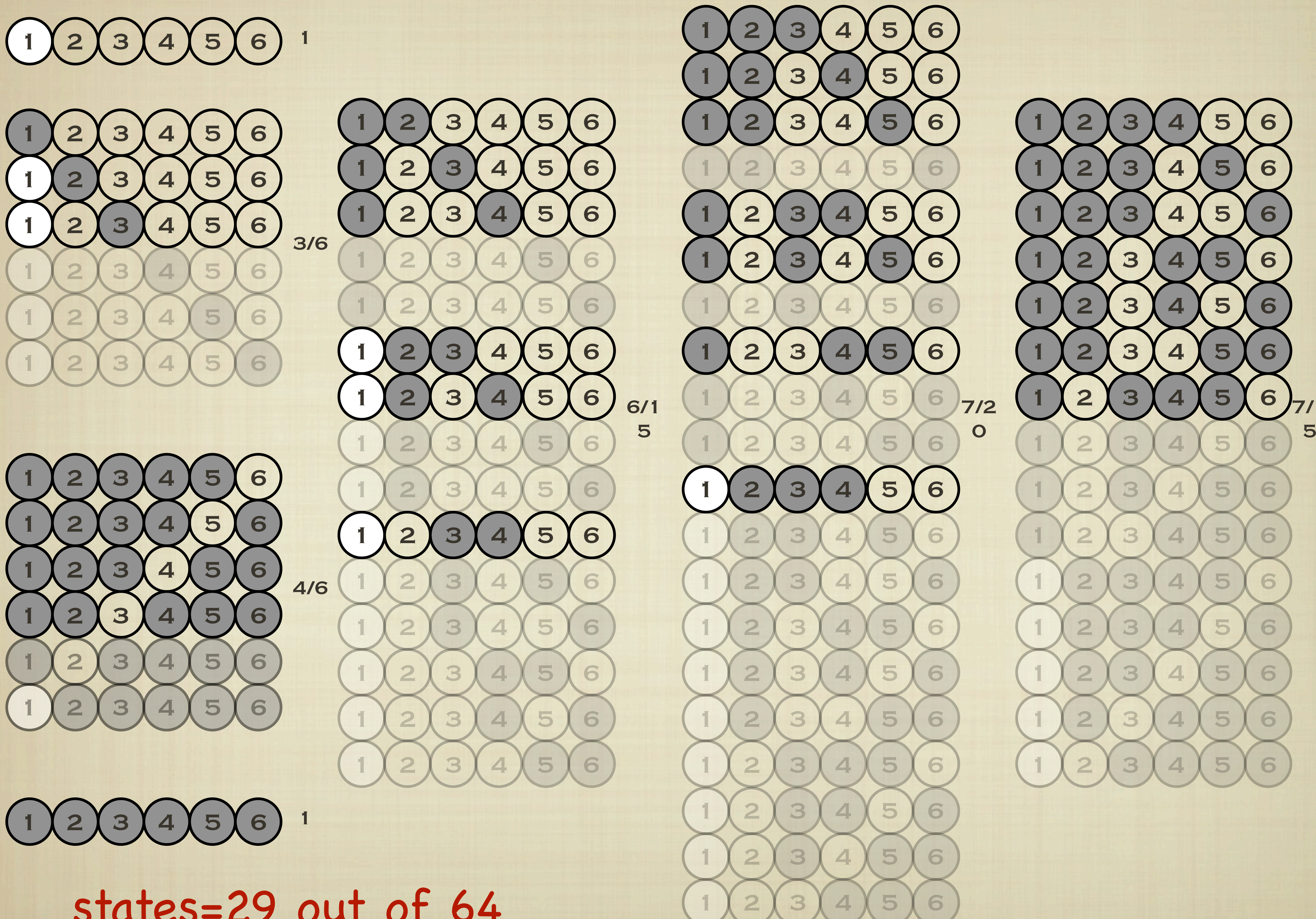
N=6, Phrase-based, S=2,WW=3



N=6, Phrase-based, S=2,WW=3



N=6, Phrase-based, S=2,WW=3



states=29 out of 64

Chart-based Decoder

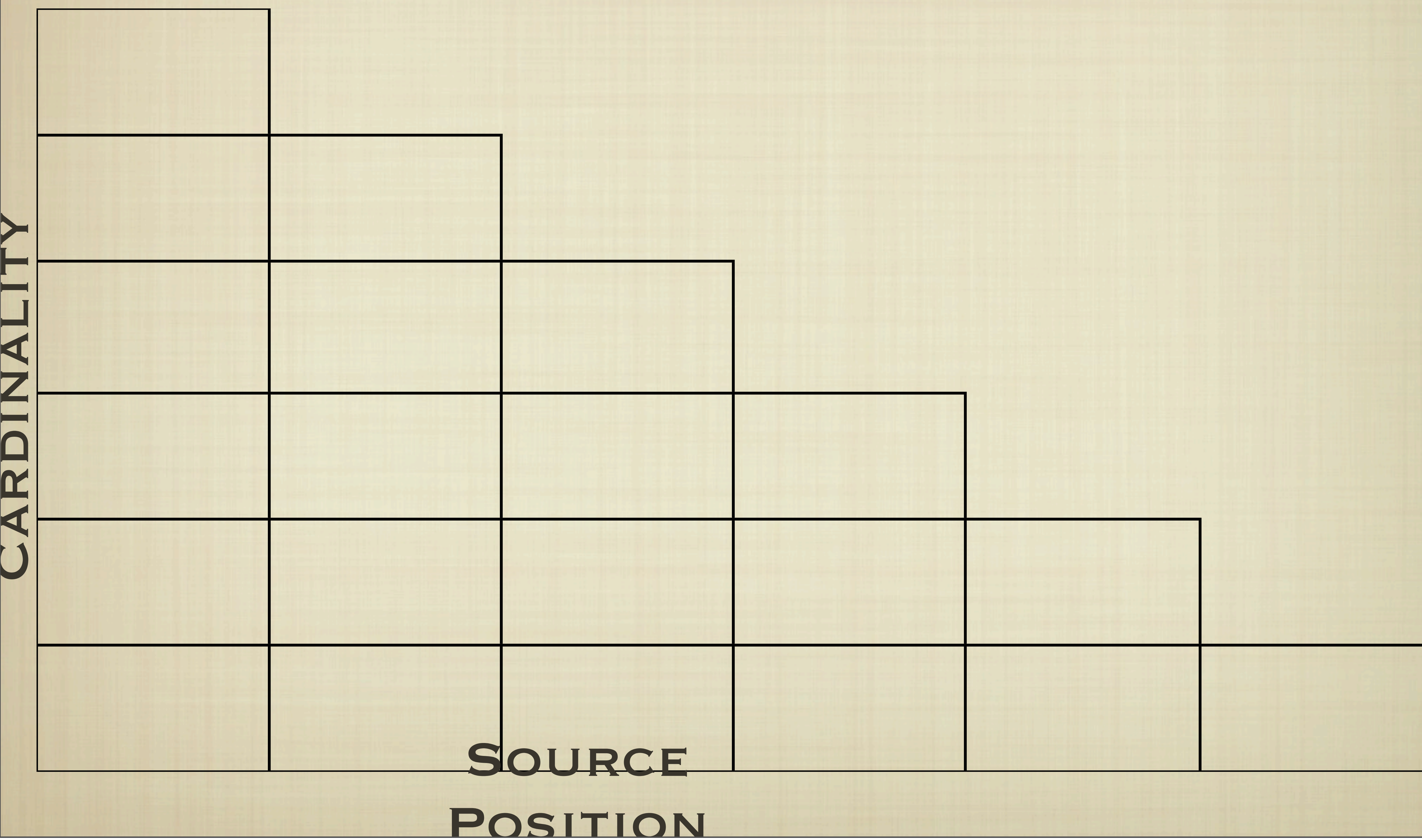


Chart-based Decoder

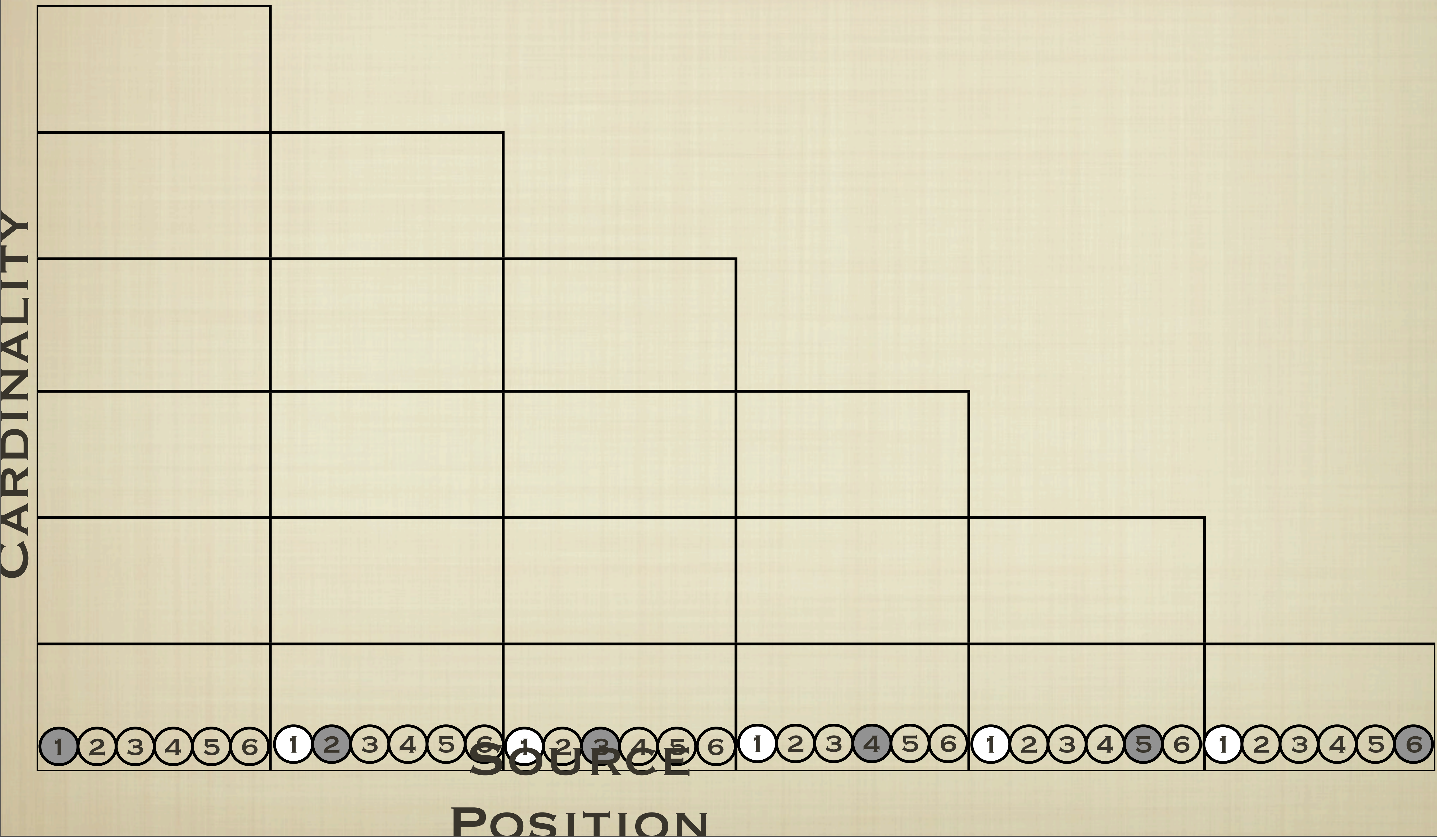


Chart-based Decoder

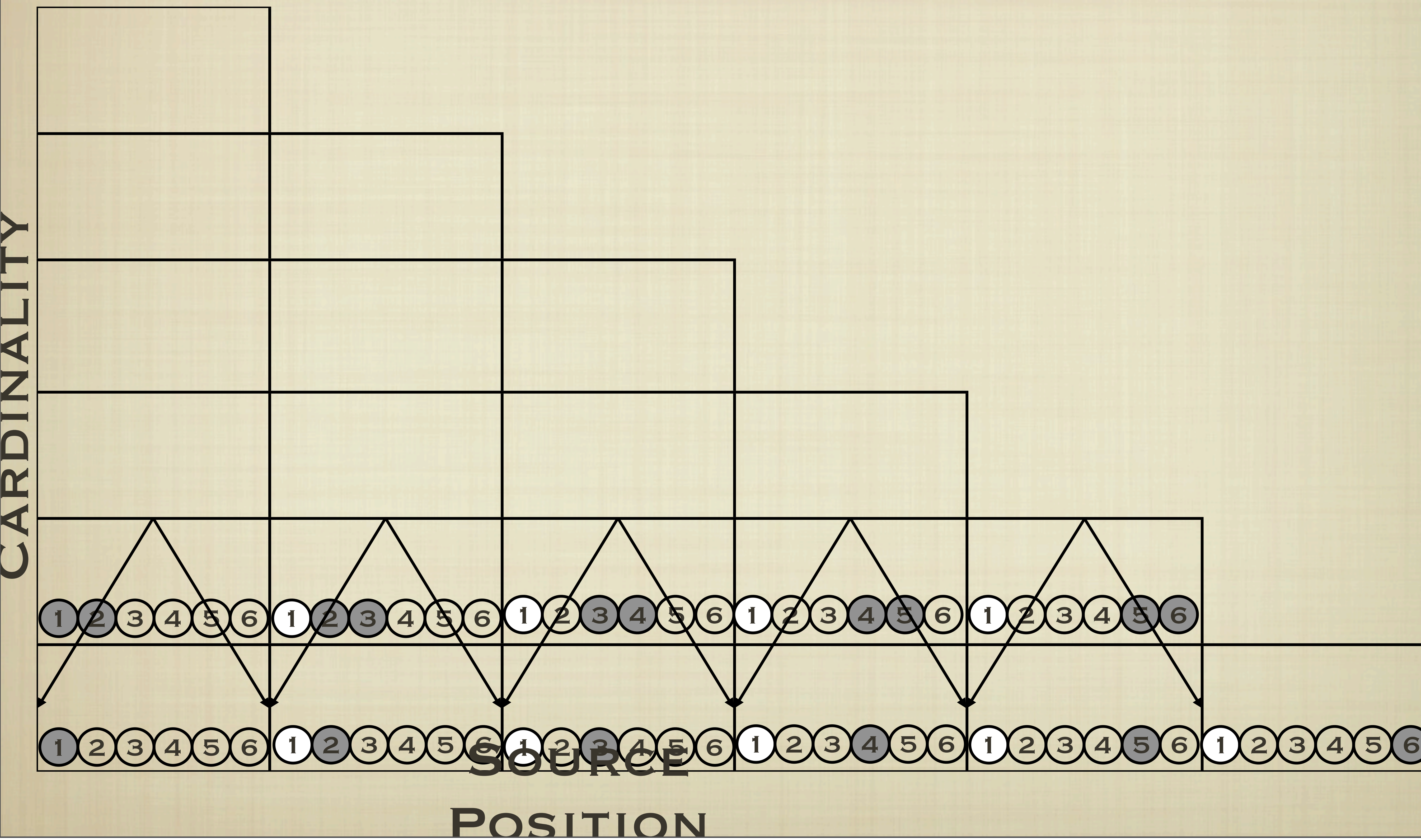


Chart-based Decoder

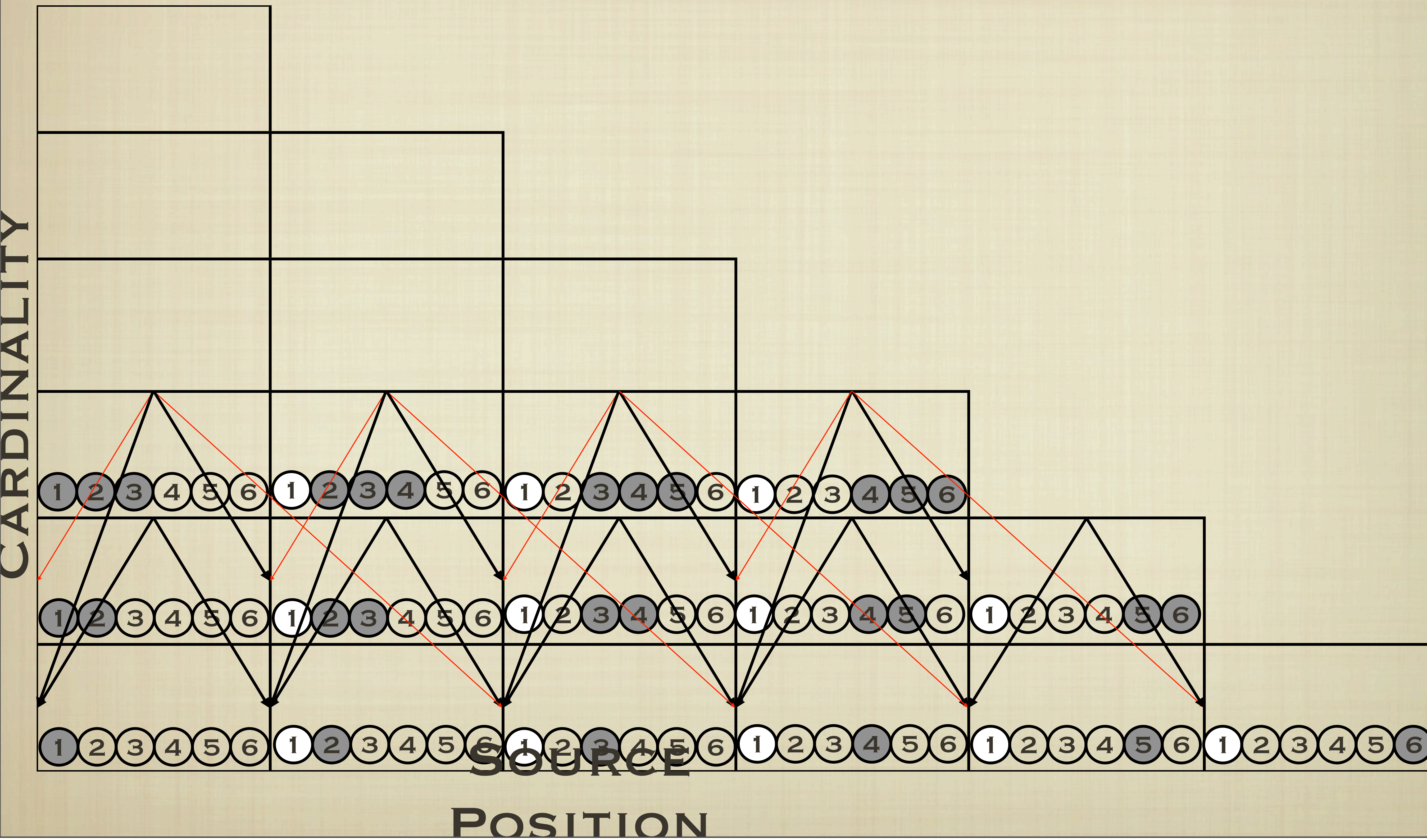


Chart-based Decoder

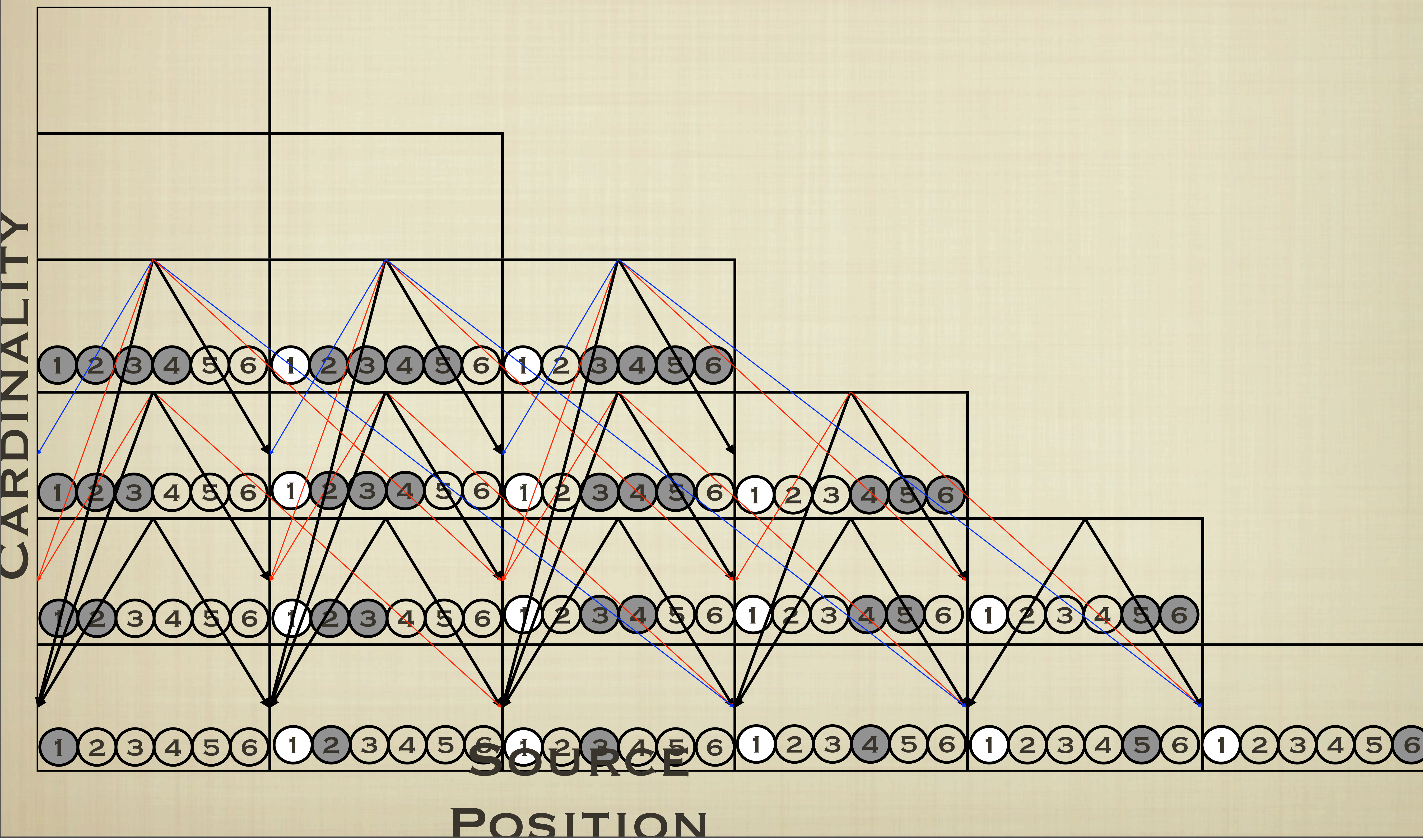


Chart-based Decoder

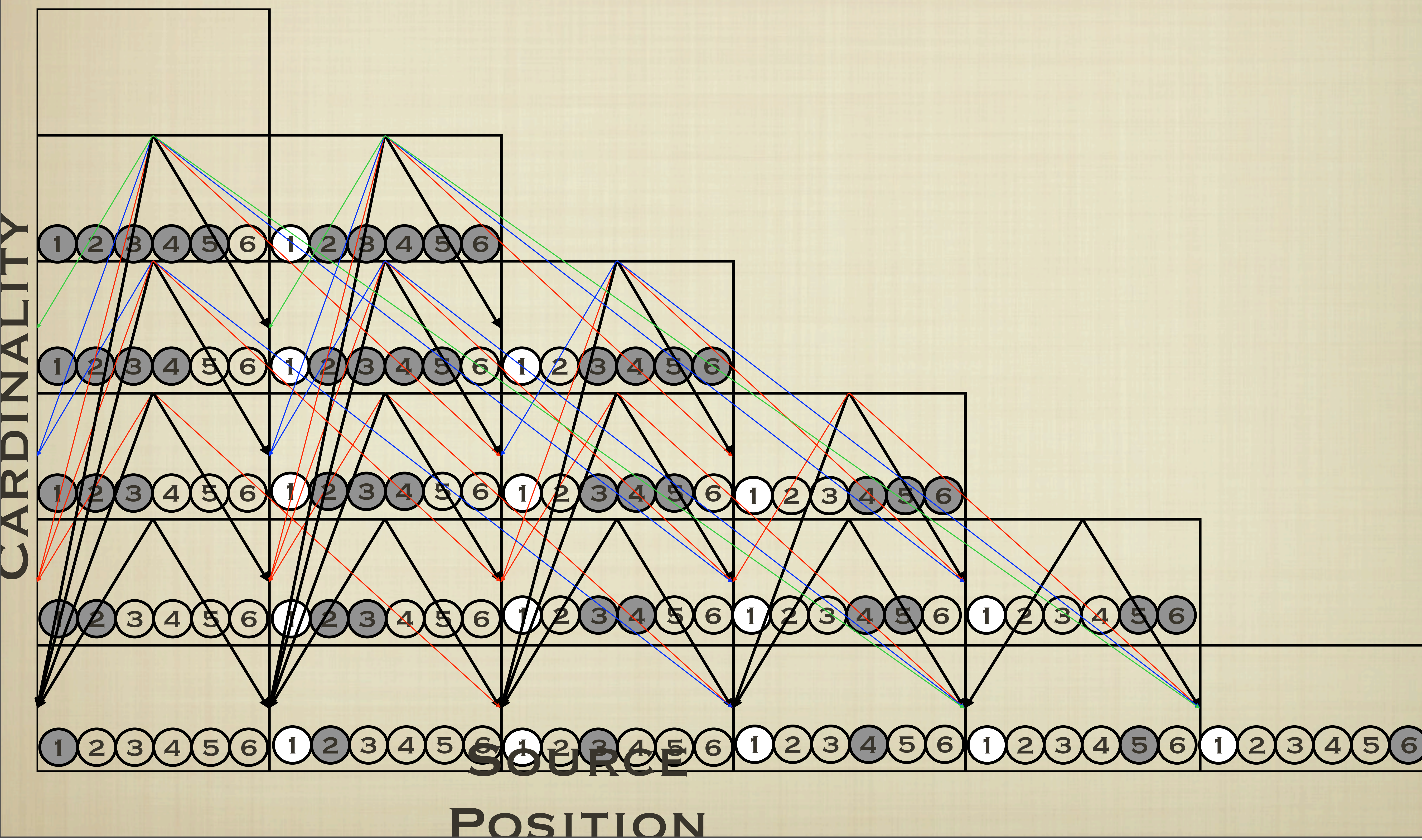
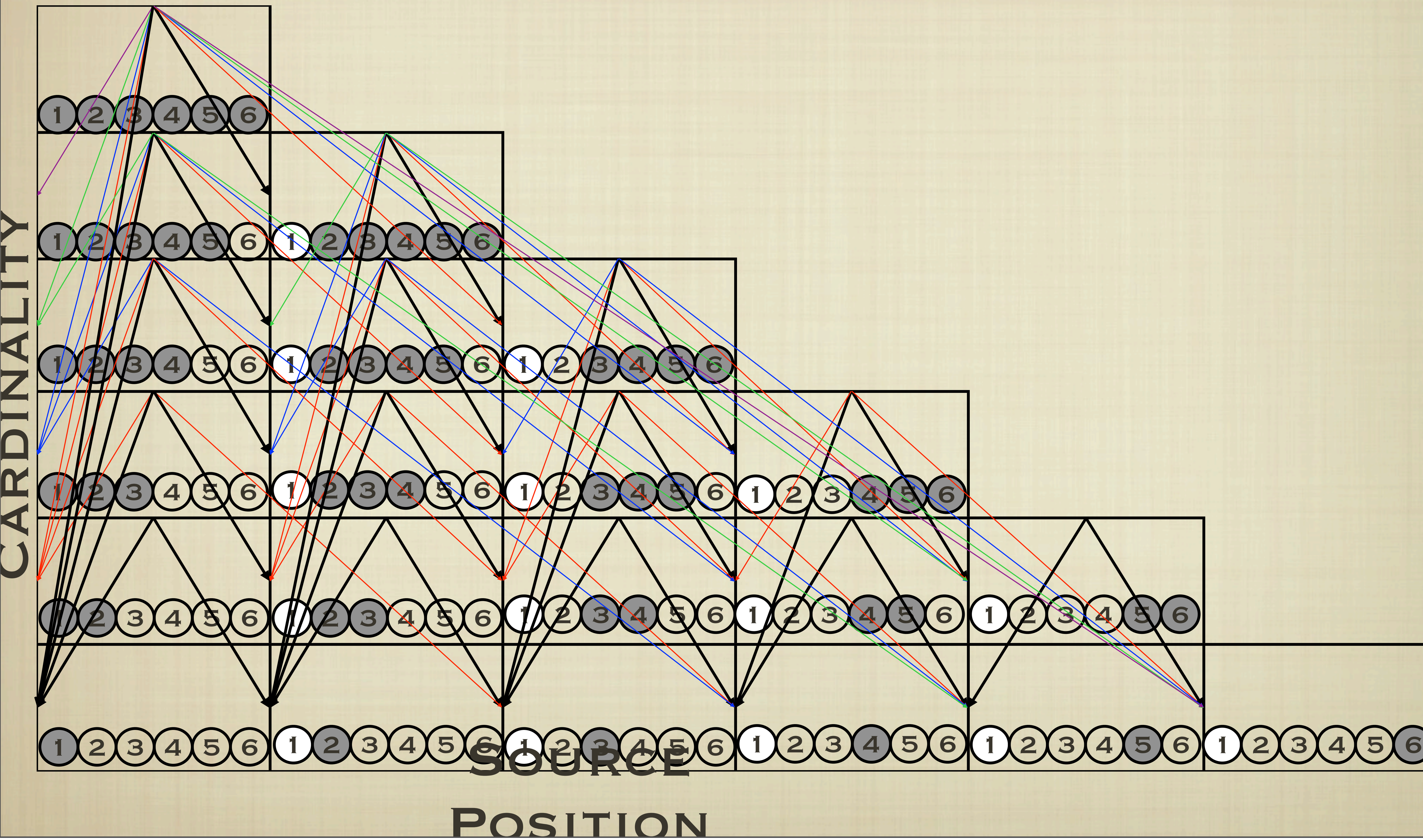
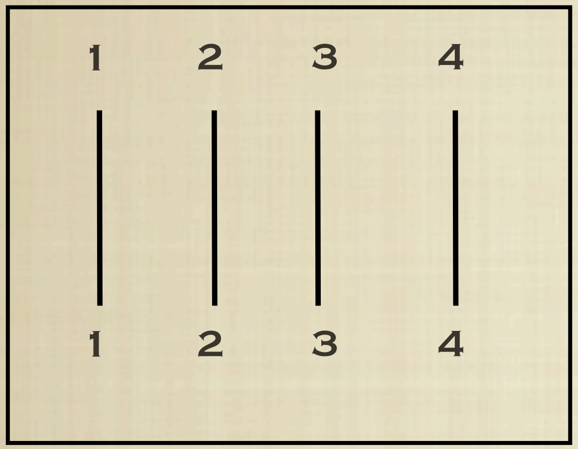


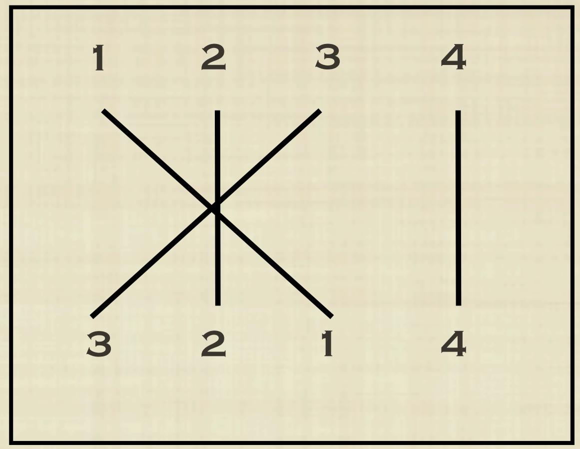
Chart-based Decoder



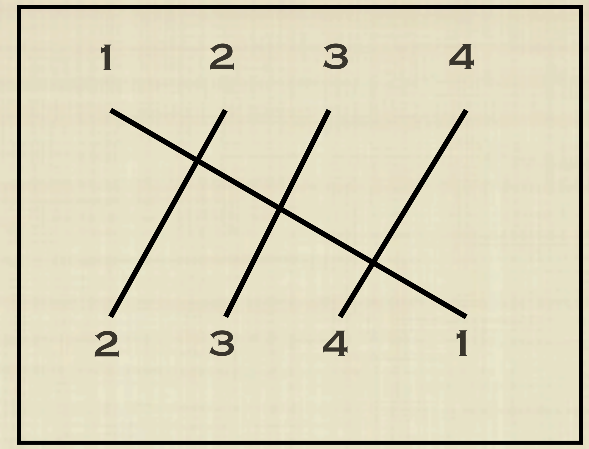
ITG Constraint



[[[1,2],3],4]

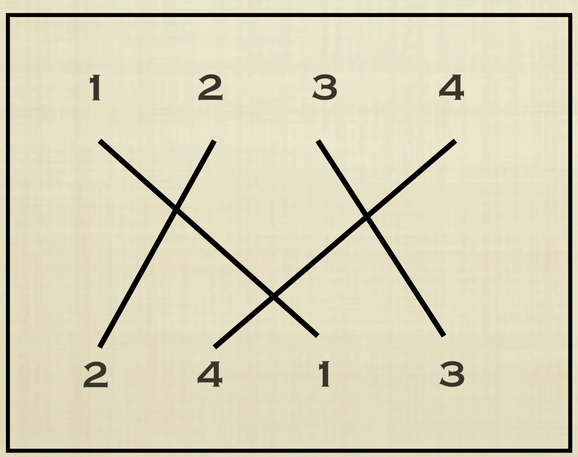


[[[3,2],1],4]

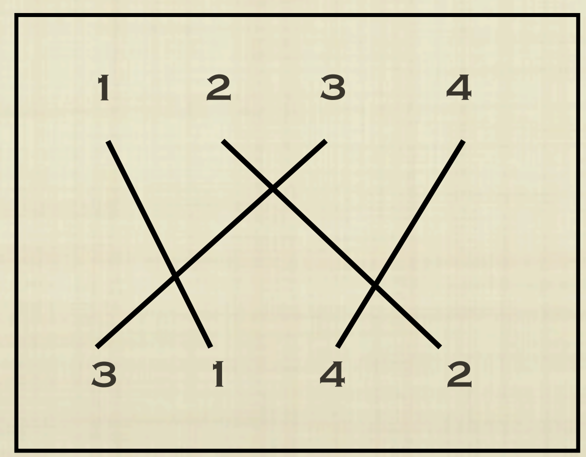


[[[2,3],4],1]

...



[2],[4],[1],[3]



[3],[1],[4],[2]

Simplified Hierarchical

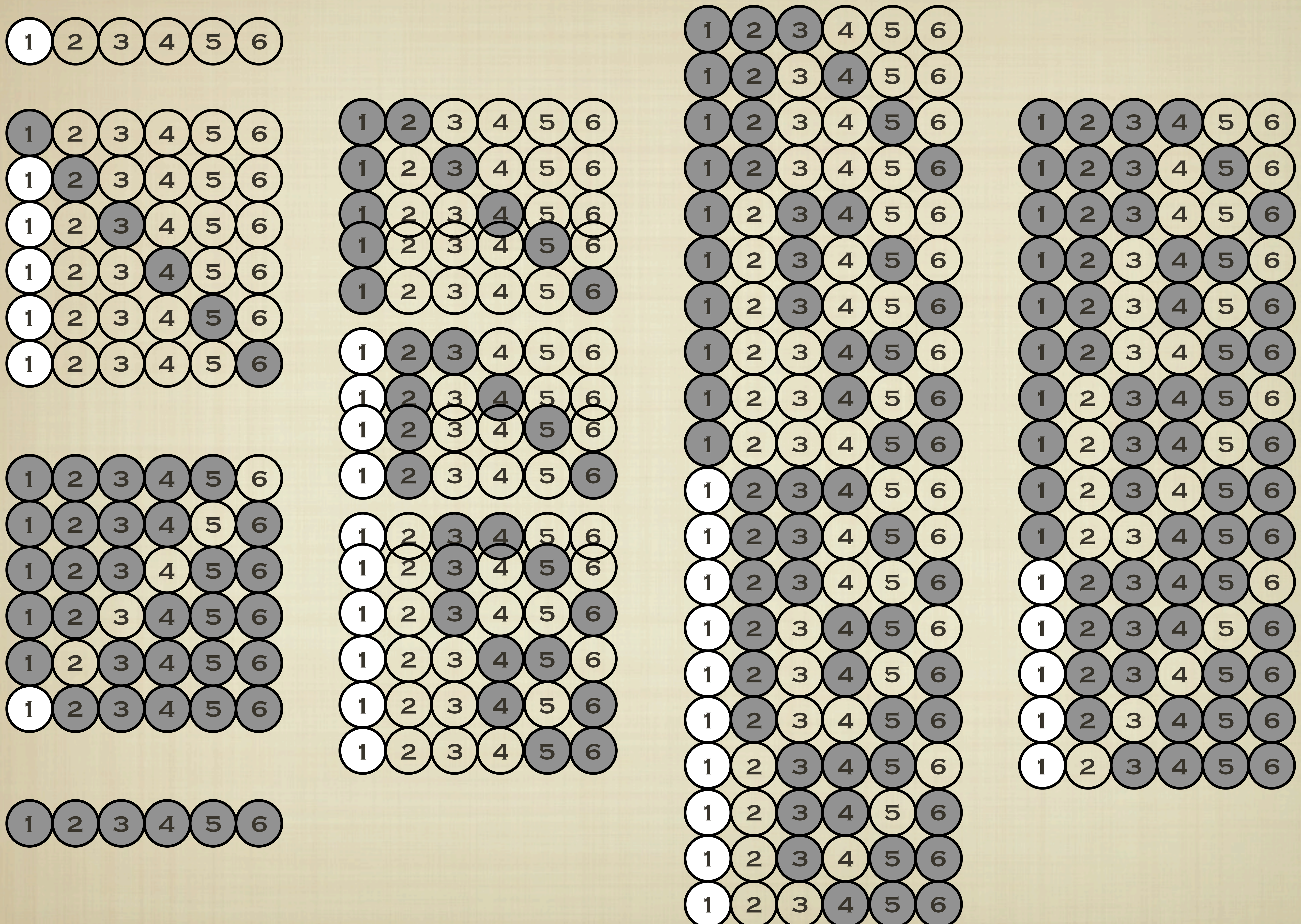
- Inversion Transduction Grammars (ITG)

Wu, 1997

X: [X1] [X2] , [X1] [X2] "glue rule"

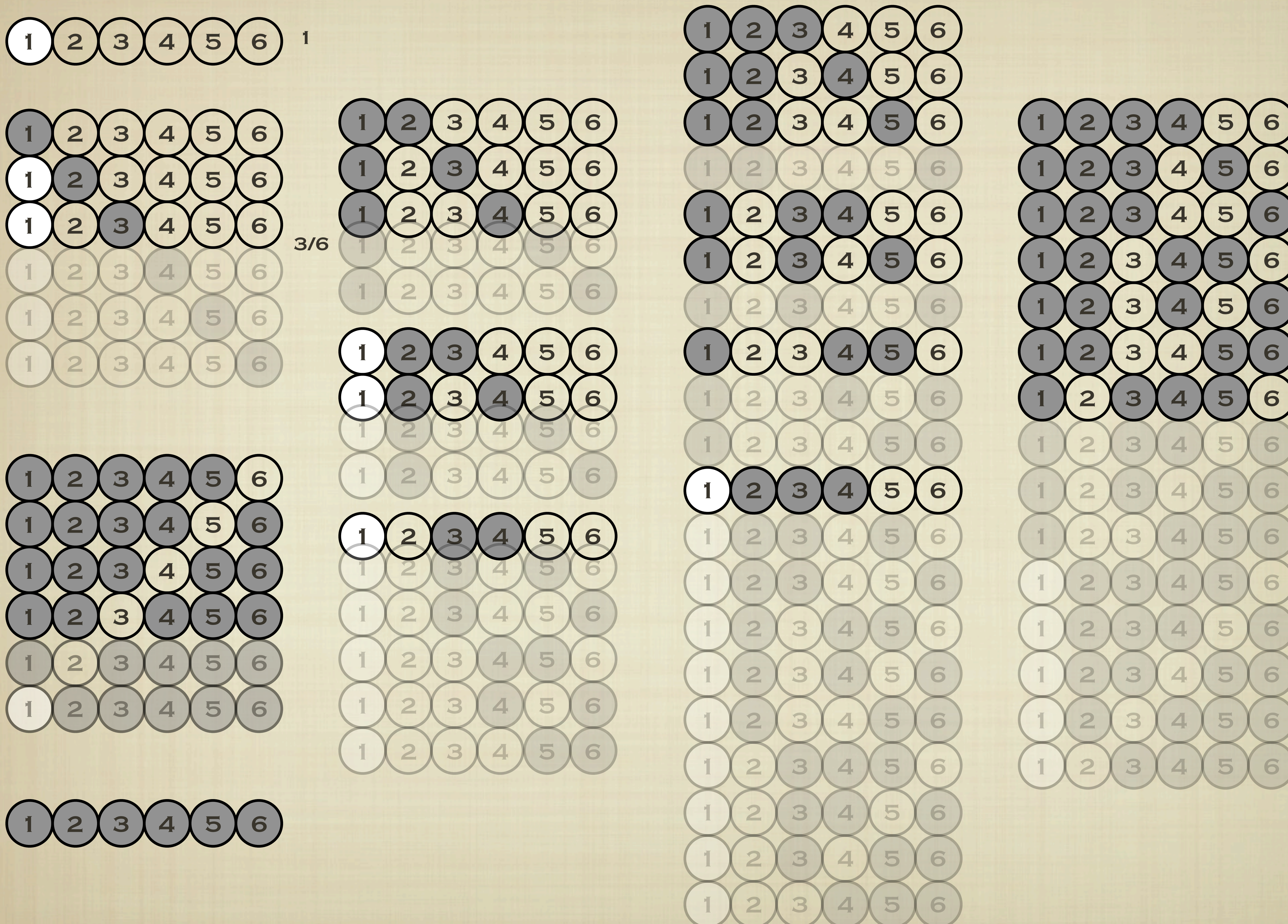
X: [X1] [X2] , [X2] [X1] "swap rule"

N=6: Hierarchical (ITG)

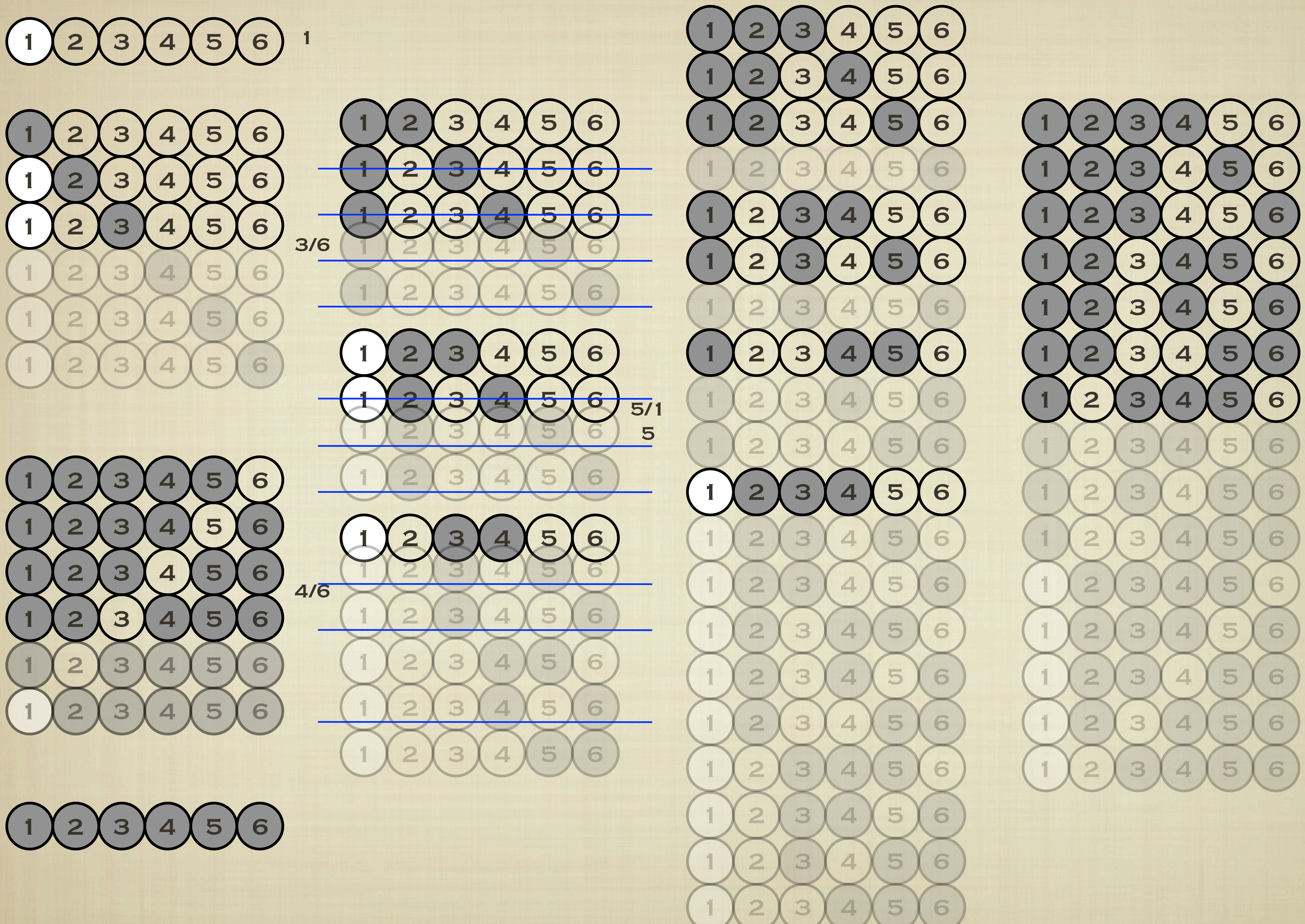


Wednesday, November 21, 12

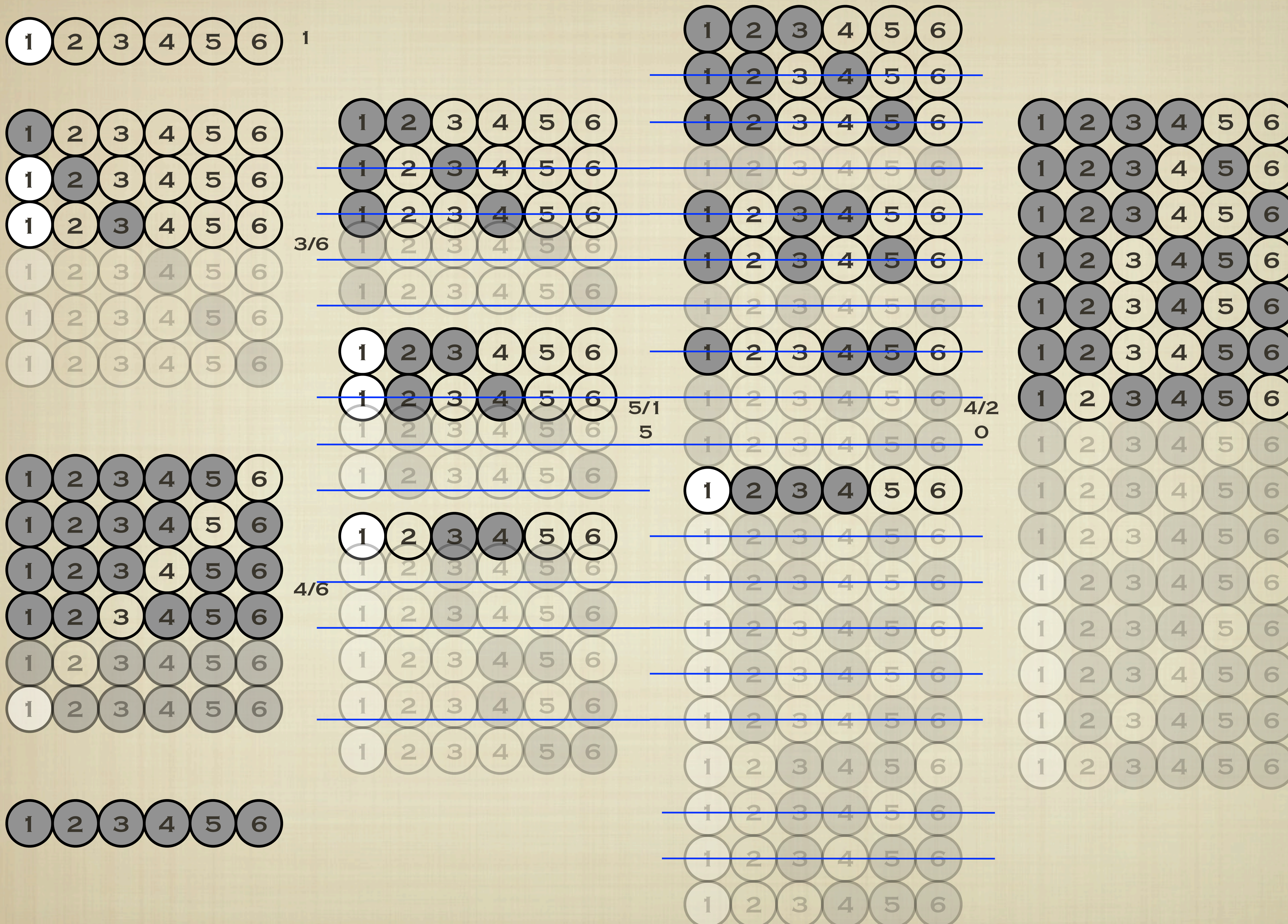
N=6: Hierarchical (ITG)



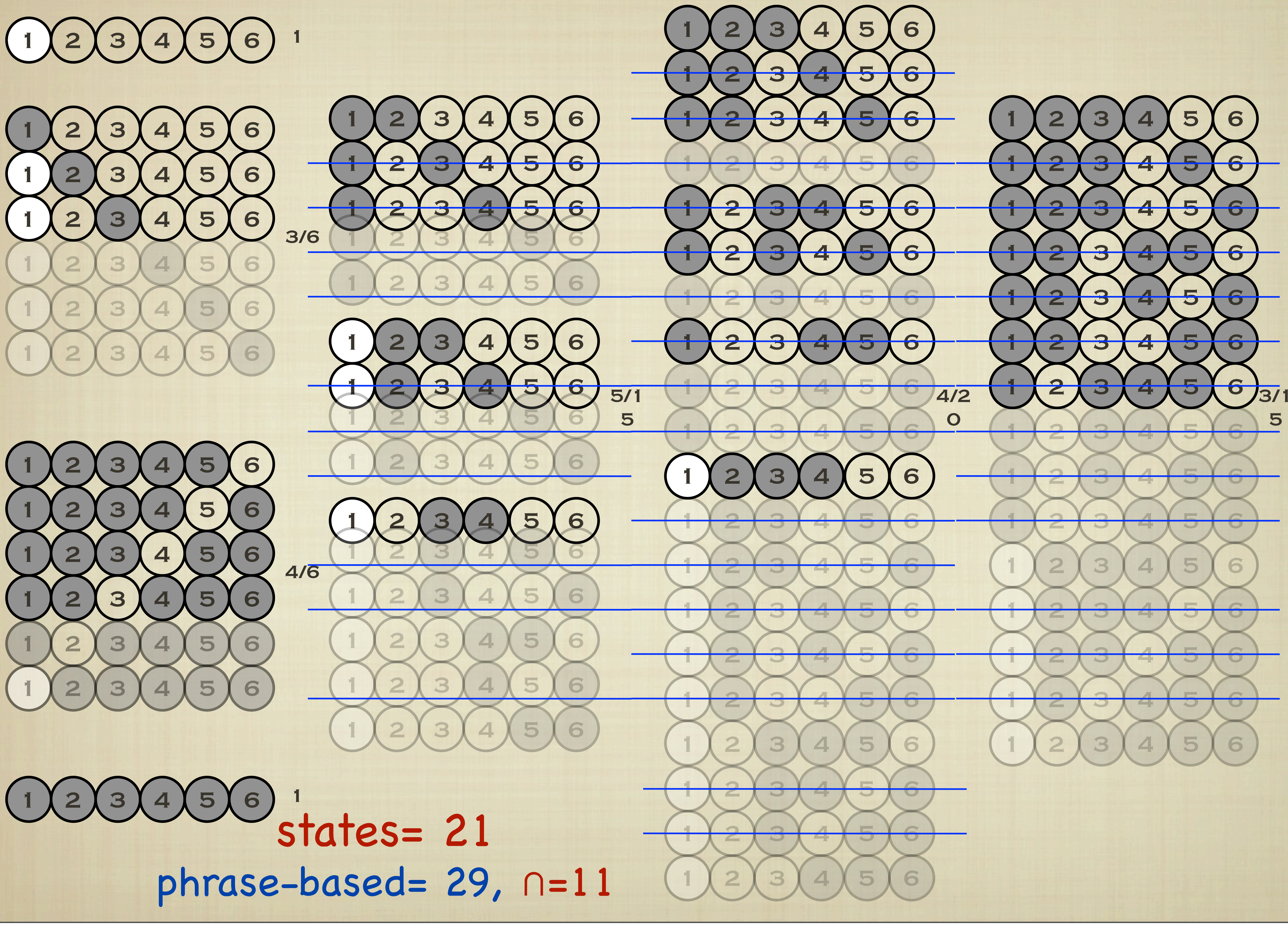
N=6: Hierarchical (ITG)



N=6: Hierarchical (ITG)



N=6: Hierarchical (ITG)



states= 21

phrase-based= 29, ∩=11

Chart Decoder Search Space:

	Sentence Length = n	No Constraints	Chart ITG Constraints
States	Unique Coverage Vectors	2^n	$\frac{n(n+1)}{2}$
Edges	Extensions	$\sum_{i=0}^{n} \text{Choose}(n,k) \times (n-i)$	$\frac{2n+n^3}{3}$
Paths	Source Order Permutations	$n!$	?

Comparison

	N=6			N=10		
	ALL	ITG	S,WW=2,3	ALL	ITG	S,WW=2,3
States	64	21	29	1024	55	57
Edges	192	72	60	5120	340	124
Paths	720	394	101	3.6e+6	2e+5	4096

Search space comparison