

The Information Bottleneck

With William Bialek, Lillian Lee and Naftali Tishby

<http://xxx.lanl.gov/pdf/physics/0004057>

What is the information content of a document?

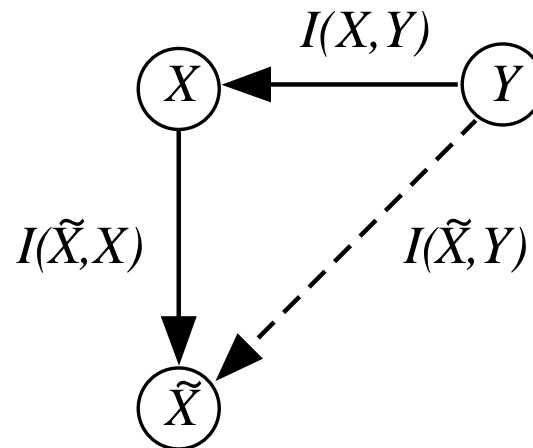
- Depends on what you want to know:
 - the exact text: *traditional information theory*
 - the topic of the document
 - who did what to whom
 - writing style
 - . . .
- Tailoring the representation to the question:
 - Quantify “information about”
 - Learn a concise representation preserving information about the question

Information-Preserving Summaries

- Observed variable X , variable of interest Y
- How much information does X have about Y :

$$I(X, Y) = \sum_x \sum_y p(x, y) \log \left[\frac{p(x, y)}{p(x)p(y)} \right]$$

- Summarize X into \tilde{X} preserving as much information about Y as possible
- Probabilistic compression rule $p(\tilde{X}|X)$



Bottleneck Solution

- Maximize $I(\tilde{X}, Y)$ at fixed compression rate $I(X, \tilde{X})$ (Lagrange multiplier β):

$$p(\tilde{x}|x) = \frac{p(\tilde{x})}{Z(\beta, x)} \exp[-\beta D(p(Y|x) || p(Y|\tilde{x}))]$$

$$p(\tilde{x}) = \sum_x p(x)p(\tilde{x}|x)$$

$$p(y|\tilde{x}) = \frac{1}{p(\tilde{x})} \sum_x p(y|x)p(\tilde{x}|x)p(x)$$

- X s with similar Y distributions map similarly to \tilde{X} : *distributional clustering*

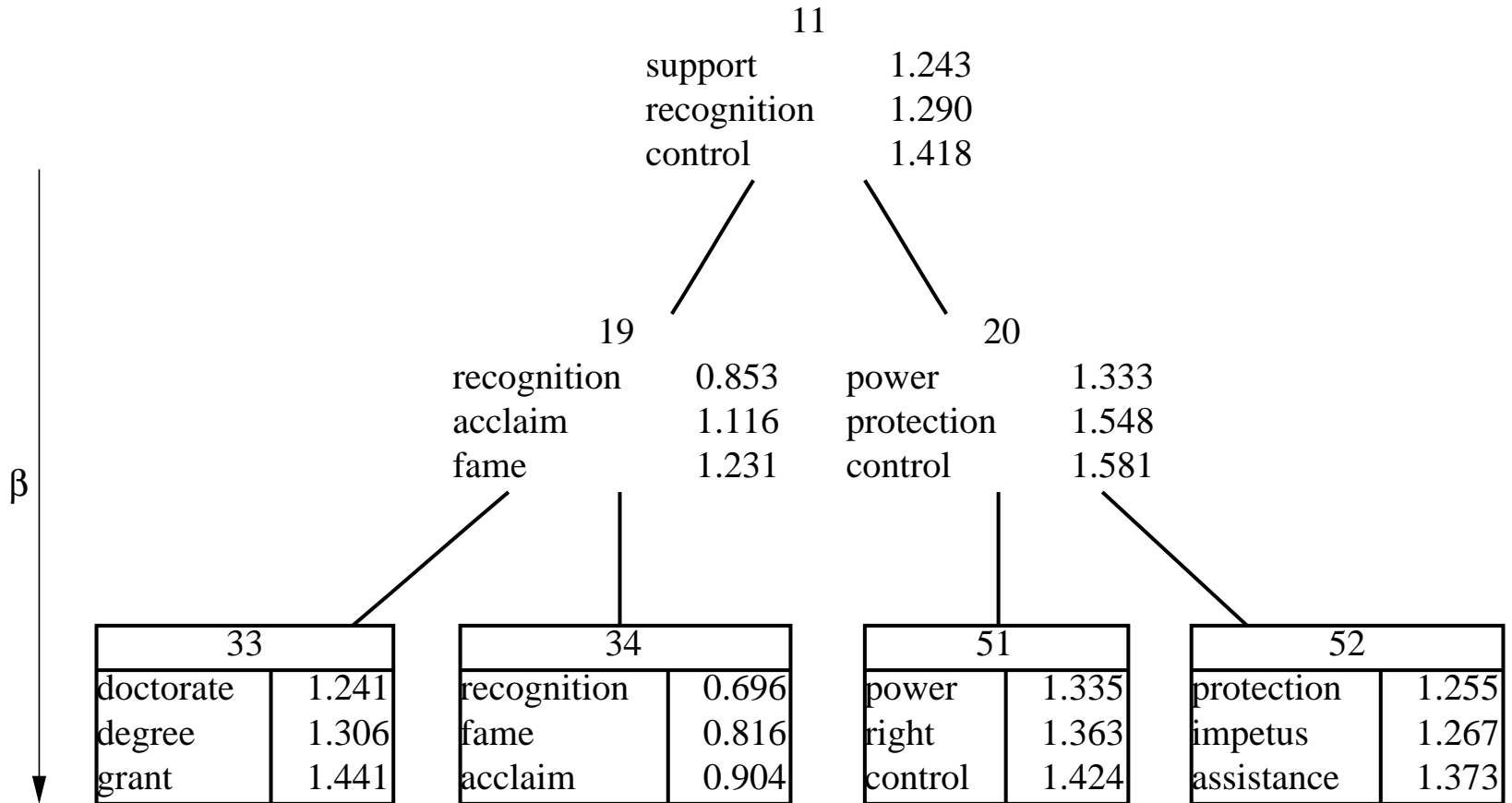
Distributional Clustering of English Words

- Group nouns by verbs that take them as direct object

X	nouns
Y	verbs
$p(Y x)$	histogram of verbs taking x as direct object
\tilde{X}	noun clusters

- Applications (Lee and Pereira; Slonim and Tishby)
 - *Smoothing*: word unseen in context \Rightarrow replace by cluster
 - *Text classification*: bag-of-words \Rightarrow bag-of-clusters
- Related work: Rooth *et al* (semantic lexicon), Hofmann (probabilistic latent semantic analysis)

Hierarchical Clustering



Information Extraction with Log-Linear Conditional Models

with

John Lafferty (CMU), Andrew McCallum (WhizBang! Labs), Dayne Freitag (Burning Glass)

- Goal: mark up sequence with *content tags*
- Problem: *overlapping dependencies on context*. For text:
 - layout
 - capitalization
 - word identity
 - *previous decisions*
- Doesn't match HMM independence assumptions
- Needs fewer states

Maximum-Entropy Markov Models

- Represent probability $P(s'|o, s)$ of new state given observation and previous state as a product of feature effects: *maximum-entropy model*

$$P(s'|s, o) = \frac{1}{Z(s, o)} \exp \left(\sum_a \lambda_{a,s,s'} f_a(o, s, s') \right)$$

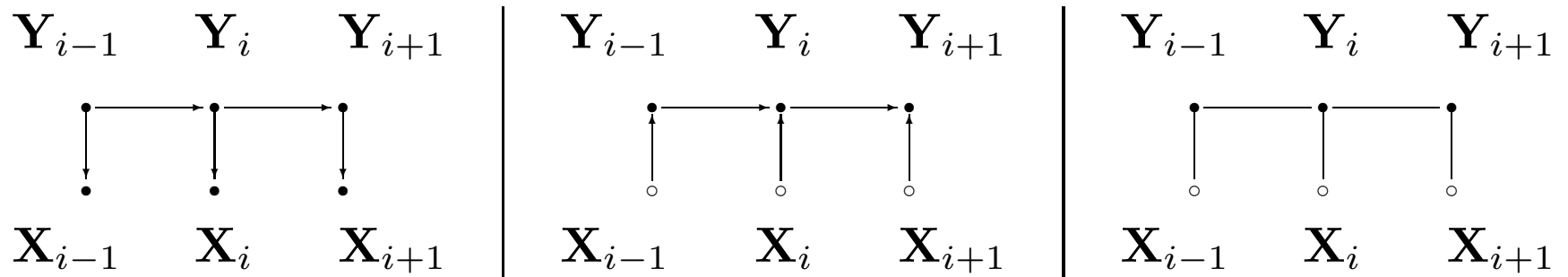
- Training algorithm:
 - Estimate the *expected* frequency of each transition: in general, *forward-backward* (partially hidden states)
 - Adjust the $\lambda_{a,s,s'}$ to *maximize* the likelihood of the observed data: *iterative scaling*

The Label Bias Problem in Conditional Models

- *Label bias*: transitions leaving a given state compete only against each other, not with all other transitions (per-state normalization)
- Conservation of score mass
- \Rightarrow bias toward states with fewer outgoing transitions

Conditional Random Fields

- Model conditional probability of whole state sequence
- Global rather than per-state normalization

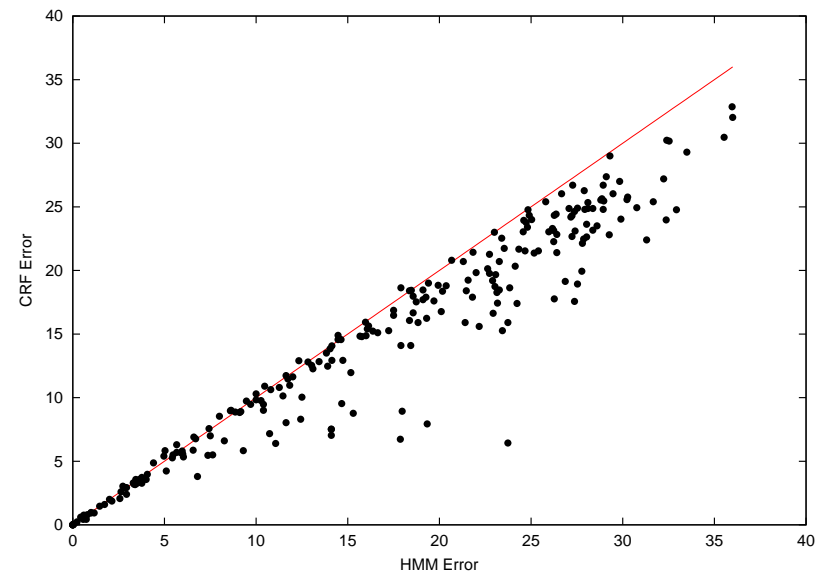
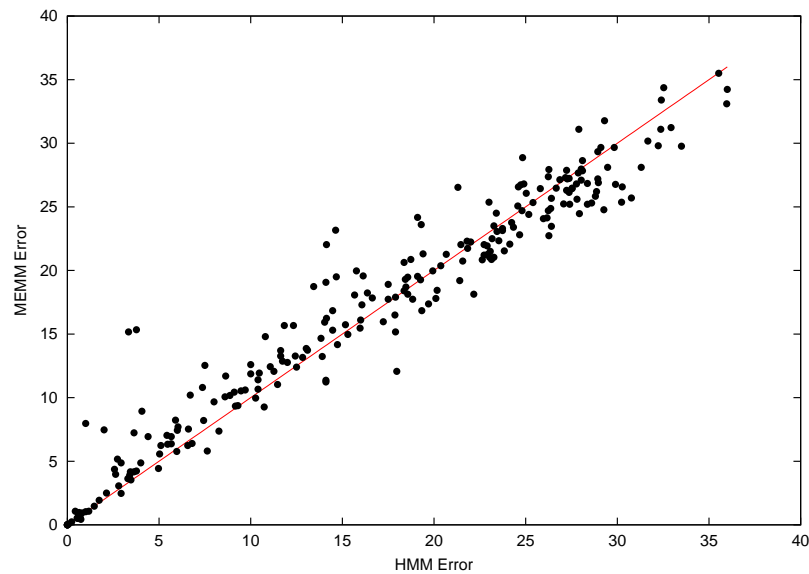


- Model form

$$p_{\theta}(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left(\sum_{e \in E, k} \lambda_k f_k(e, \mathbf{y} \mid e, \mathbf{x}) + \sum_{v \in V, k} \mu_k g_k(v, \mathbf{y} \mid v, \mathbf{x}) \right)$$

Synthetic Data Experiments

- Data generated by mixing *first* and *second* order HMMs
- Modeled by *first*-order HMM, MEMM, and CRF (without contextual or overlapping features), representing common modeling limitations



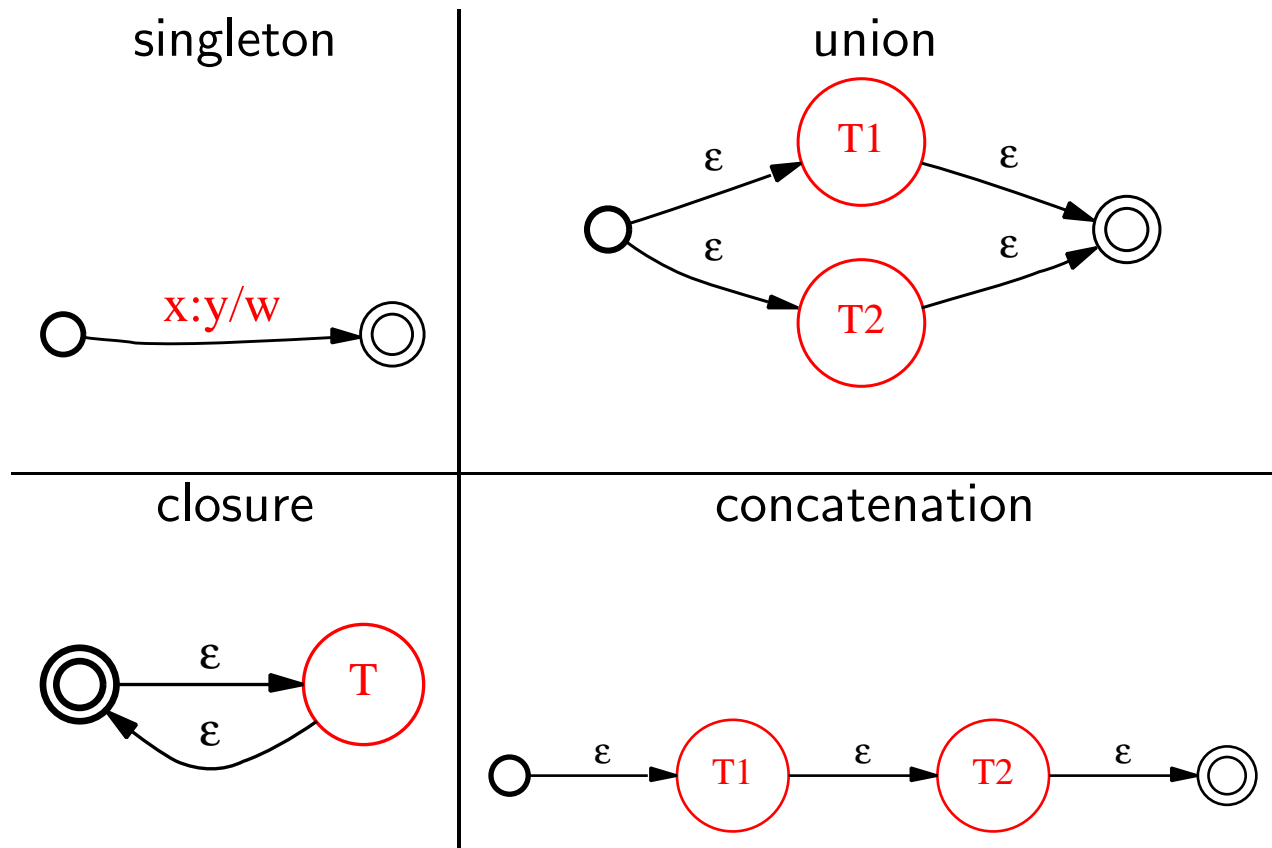
Weighted Finite-State Transduction

with

Merhyar Mohri and Michael Riley (AT&T Labs)

- *Composition* of finite-state transformations
- Common generalization of finite-state string edits, HMMs, probabilistic automata
- General *network optimization* methods: weighted transducer ϵ -removal, determinization, minimization
- Beyond probabilistic automata: multiplicity, $(-\log)$ probabilities, edit costs, . . . just need to satisfy simple algebraic properties

Rational Operations on Weighted Transducers

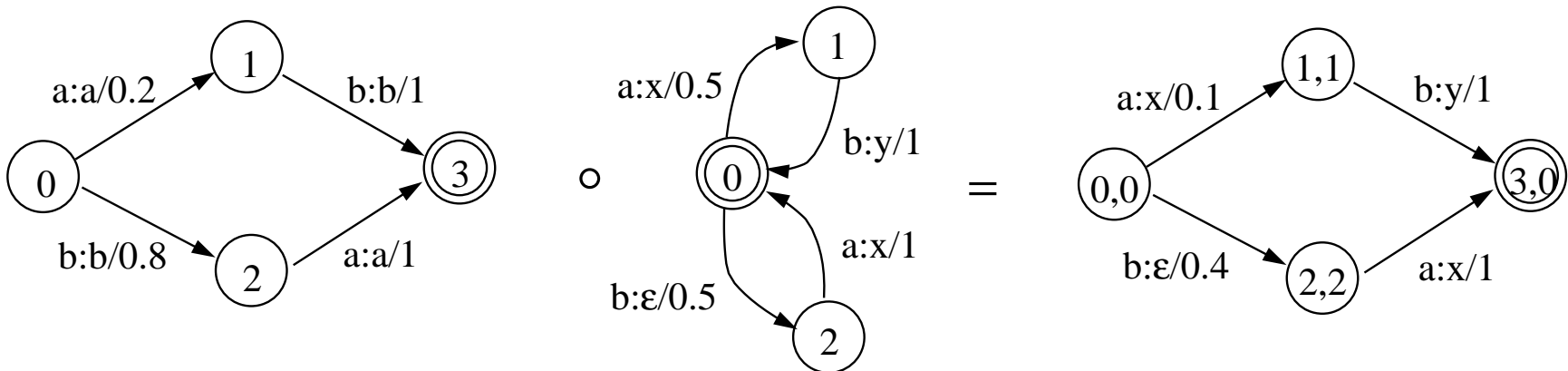


Composition of Weighted Transducers

- Generalization of finite automata intersection
- Computation rule (ϵ -free case):

$$\frac{s \xrightarrow{a:b/u} s' \quad t \xrightarrow{b:c/v} t'}{(s, t) \xrightarrow{a:c/(u \otimes v)} (s', t')}$$

- Goal-directed execution

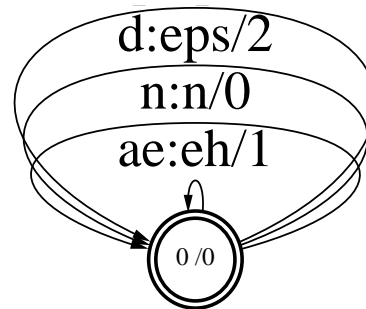


Simple Example: Weighted String Edit Distance

- Task-specific *symbol edit weights*:

a_i	b_j	$w(a_i, b_j)$	edit
ae	eh	1	substitution
d	ϵ	3	deletion
ϵ	pr	1	insertion

- As a weighted transducer:



T.fst

- More generally: multi-state transducers for context-dependent edit costs (eg. pronunciation modeling)

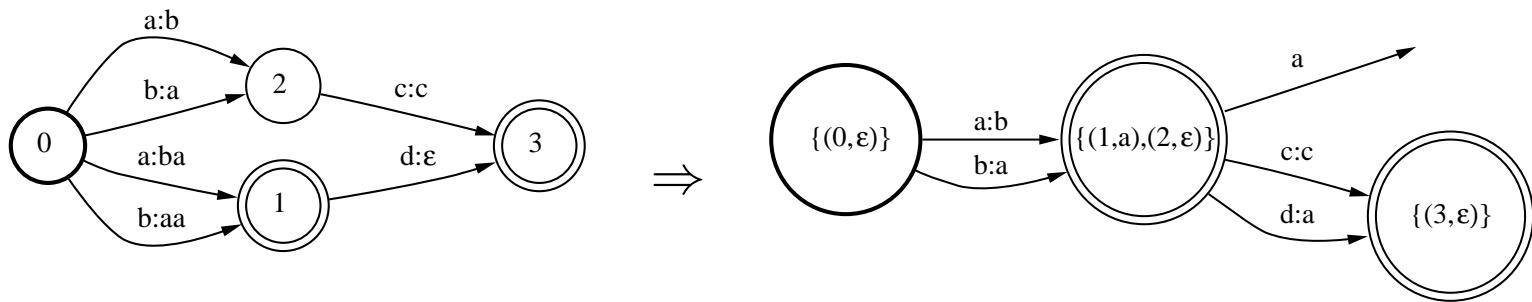
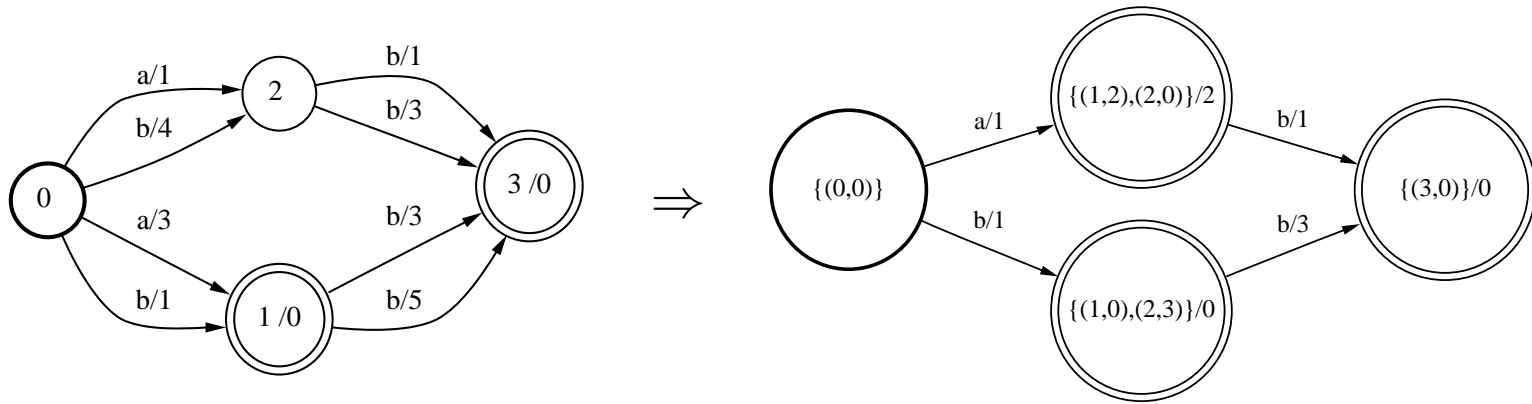
Meaning-Preserving Optimizations

- Generalizations of standard algorithms
 - *ϵ -removal*
 - *determinization*
 - *minimization*
- Weights and outputs make things subtler:
 - determinization not always possible (subsequentiality condition)
 - technical conditions on weights
- Analogies with code optimization
- Key to high-performance, large-vocabulary speech recognition

Determinization

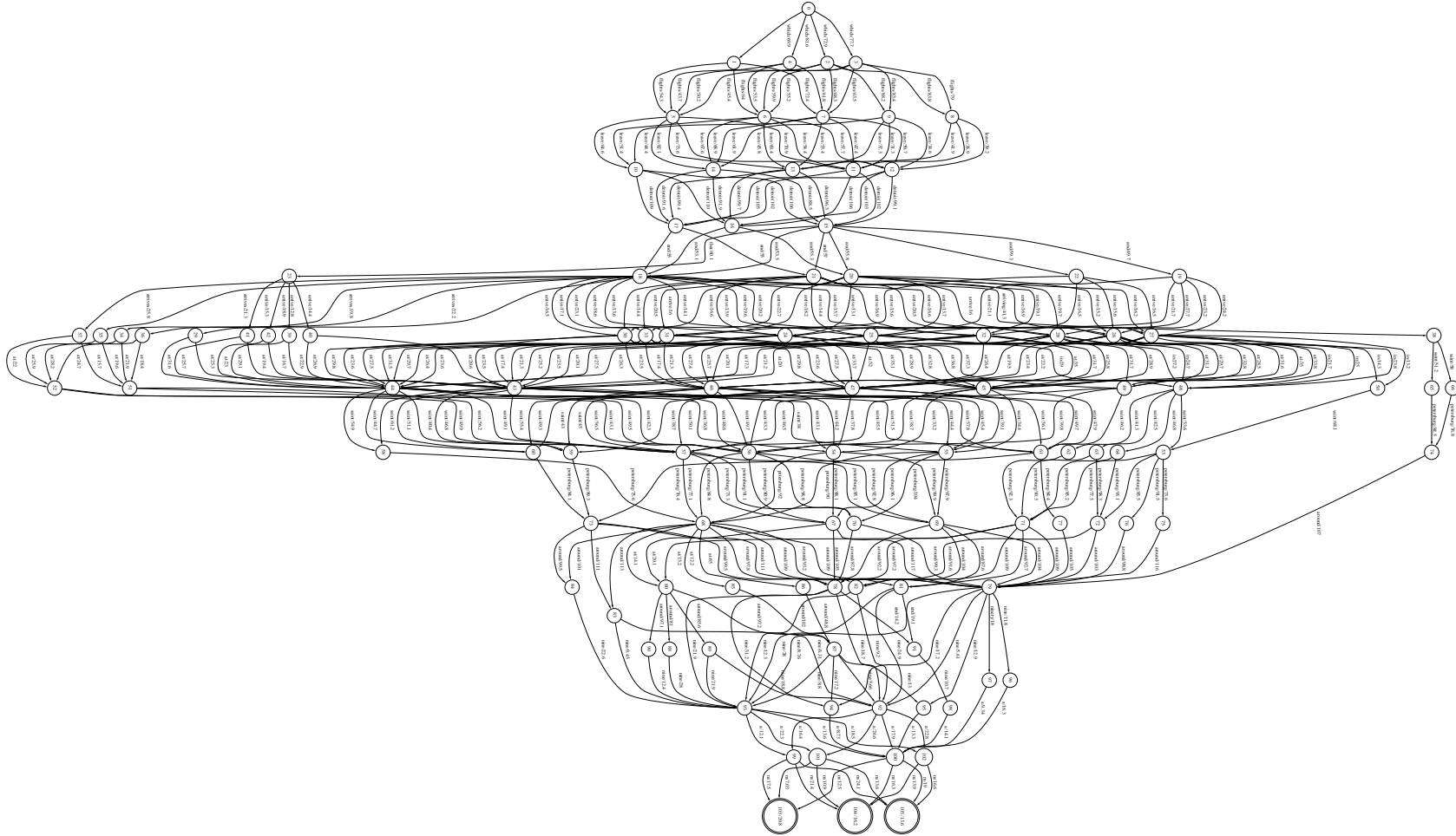
- Generalize classic subset construction
- Main ideas:
 - sets of *state-output* pairs
 - *greatest common output*
- Applications: reduce automaton size, speed up composition

Determinization Examples



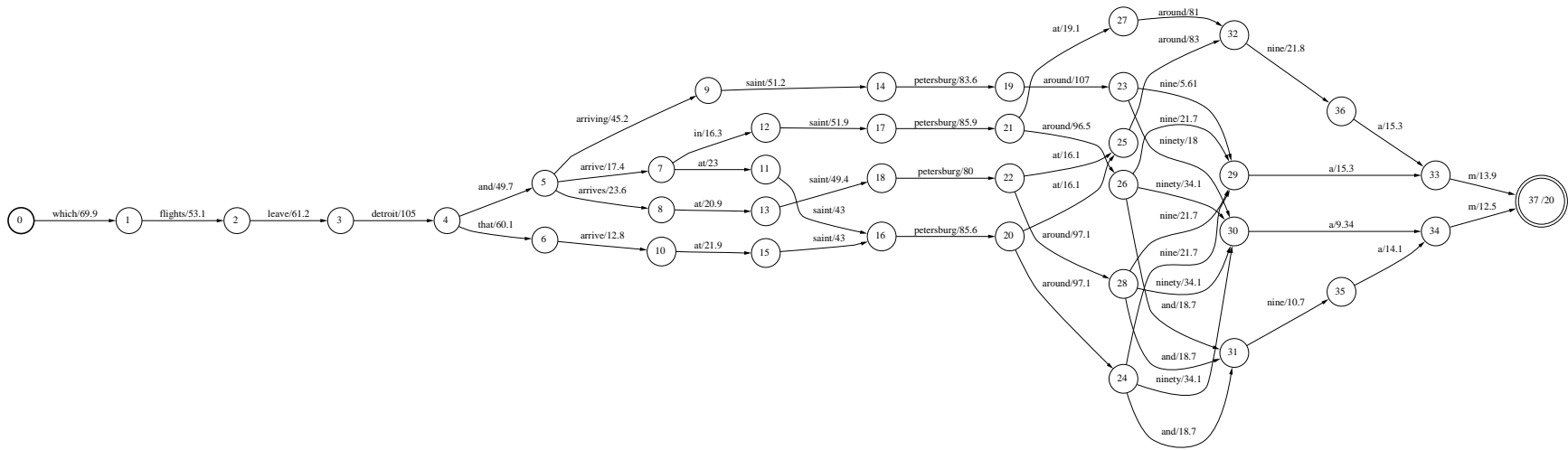
Weighted Determinization: Input

Word lattice: *106 states, 359 transitions, 82,977,532 paths.*



Weighted Determinization: Output

Determinized word lattice: *38 states, 51 transitions, 18 paths.*



d

And minimized ...

