WFST解码器

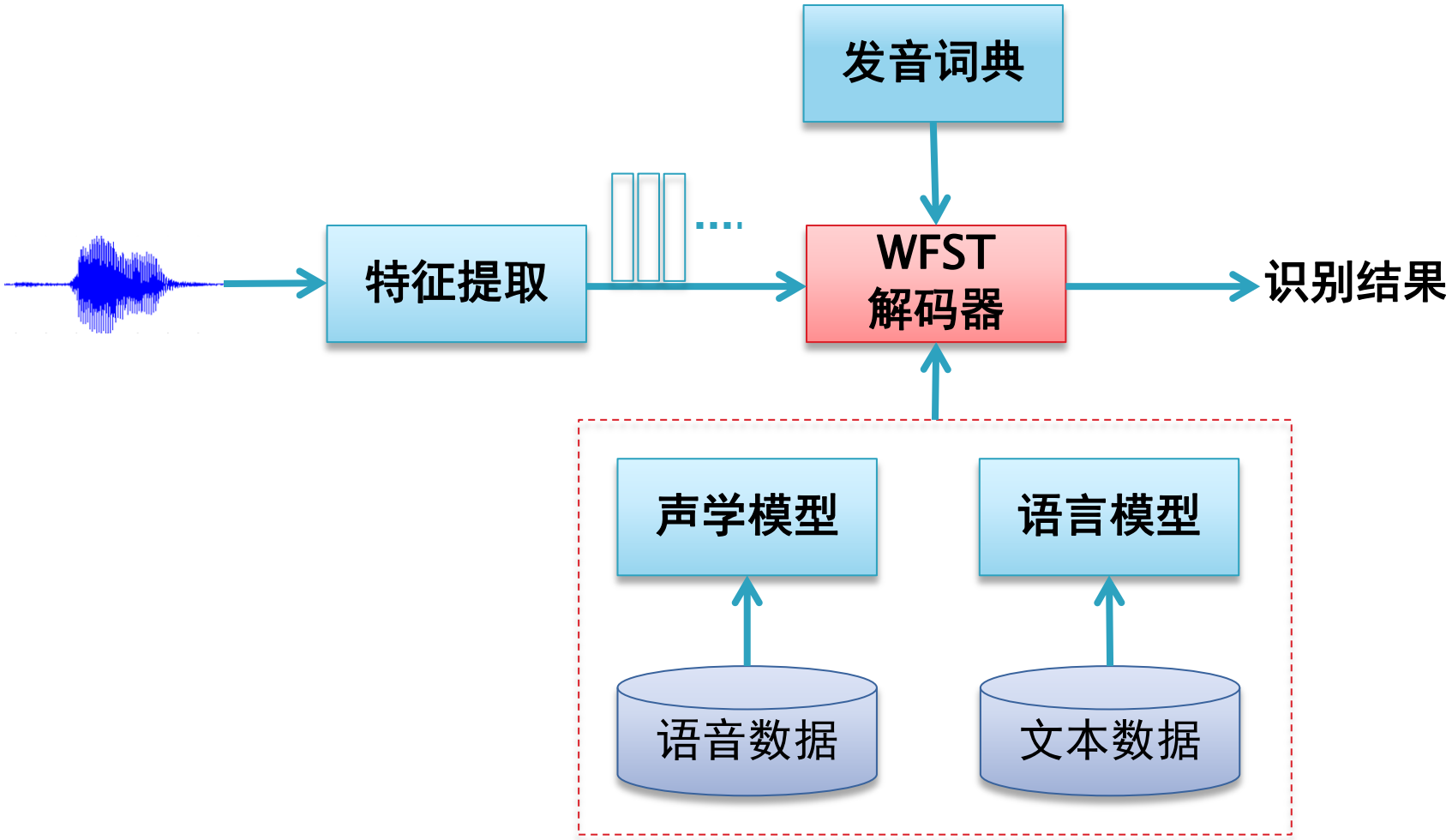
洪青阳 副教授

厦门大学信息科学与技术学院
qyhong@xmu.edu.cn

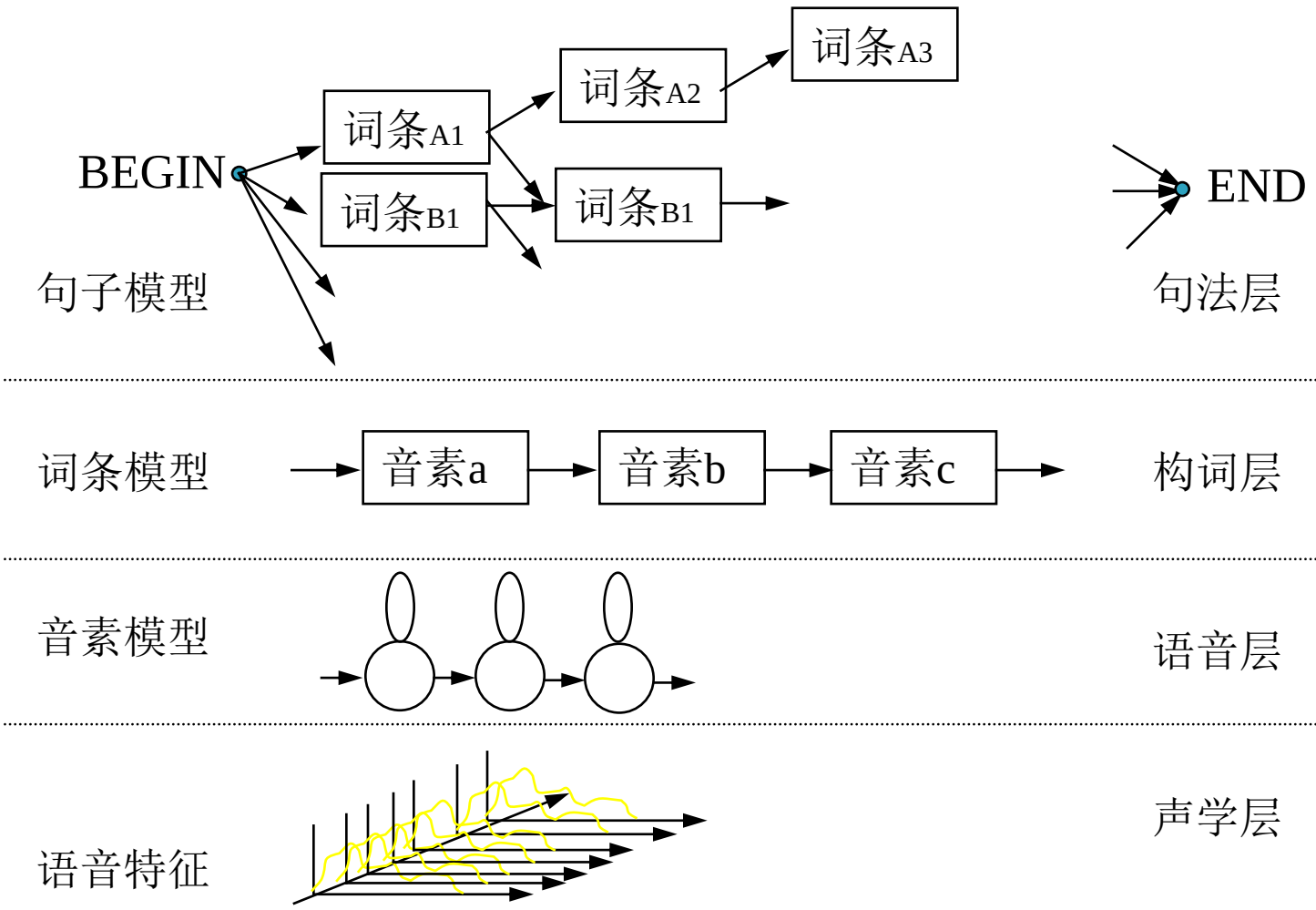
要点

- 语音识别框架
- 动态解码网络
- 静态解码网络
- WFST
- HCLG
- WFST解码
- Lattice

语音识别框架



解码过程


词条A3
词条A2
词条A1
BEGIN
词条B1
词条B1
END
句子模型
句法层
词条模型
音素a
音素b
音素c
构词层
音素模型
语音层
语音特征
声学层

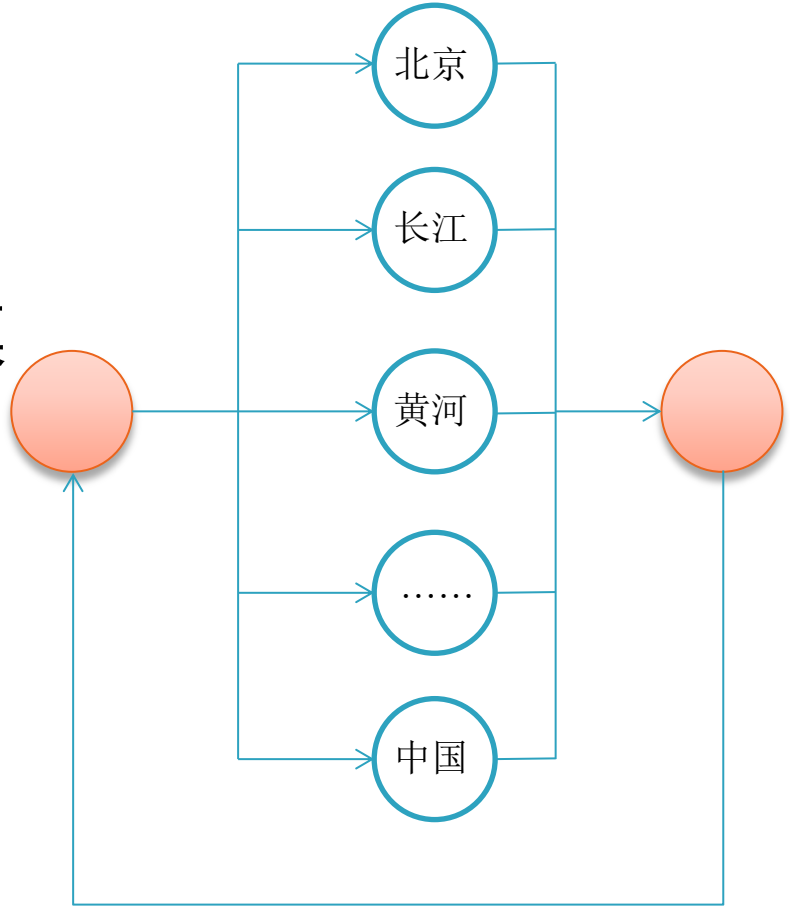
搜索空间

- **静态编译**：是把所有知识源统一编译在一个状态网络中，在解码过程中，根据节点间的转移权重获得概率信息。

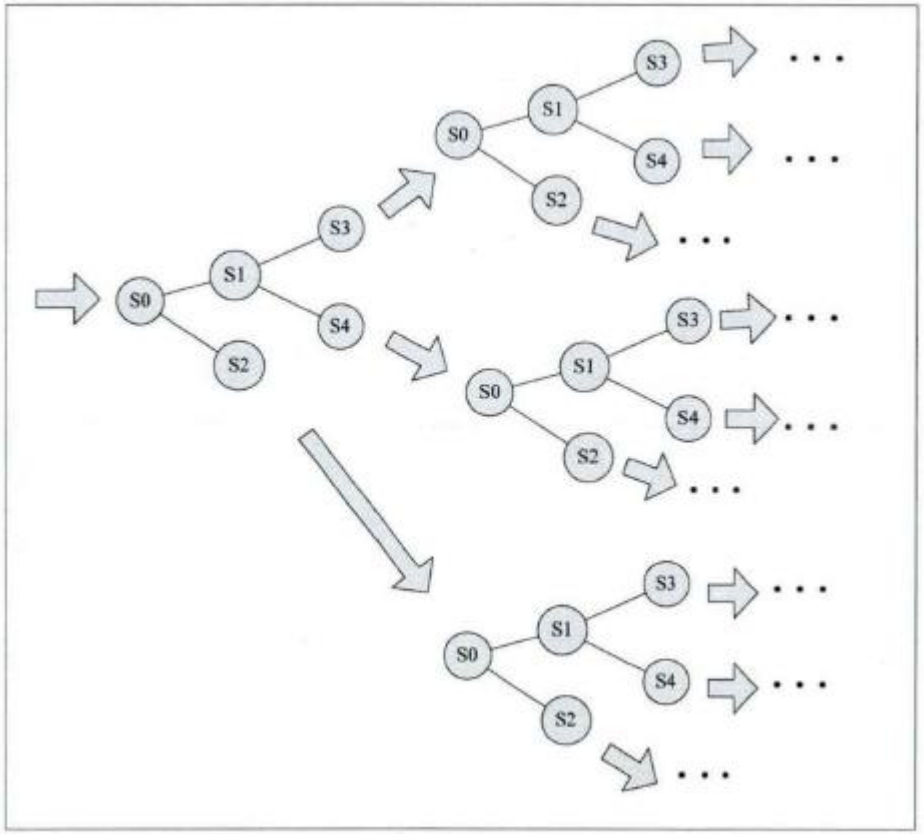
- **动态编译**：只是预先将发音词典编译成状态网络构成搜索空间，其他知识源在解码过程中根据活跃路径上携带的历史信息动态集成。

动态解码网络

- 以所有词的并列为解码网络，支持回跳循环
- 循环跳回的时候，加入语言模型概率



动态解码网络



其他知识源在解码过程中根据活跃路径上携带的历史信息动态集成。

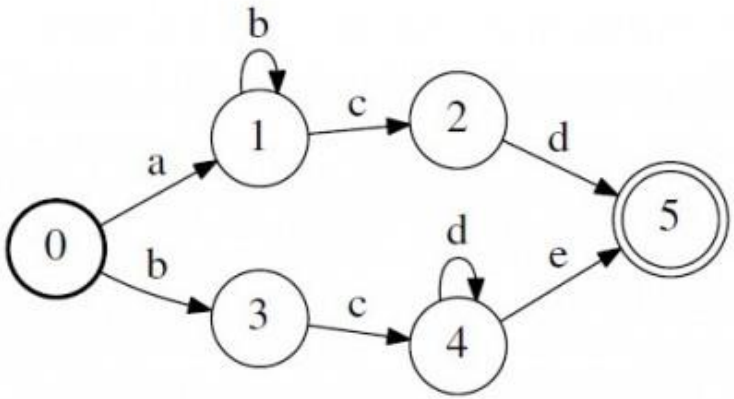
静态解码网络

- 加权有限状态转换器(WFST)，最早由AT&T的Mohri提出，现已成为大词汇量连续语音识别(LVCSR)系统的最高效的解码算法。

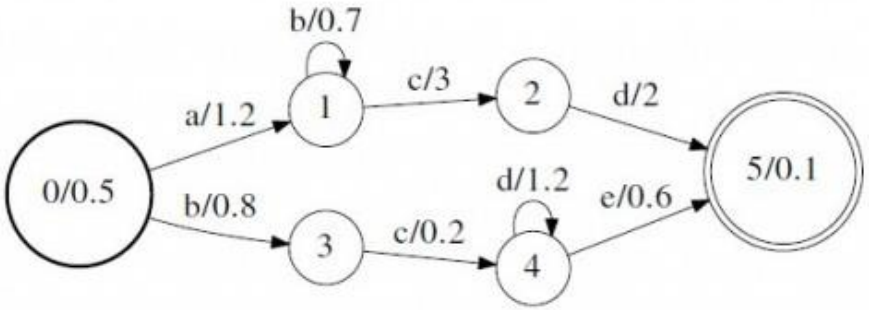
- 与传统动态网络解码相比，基于WFST的识别系统在识别之前产生识别用的静态解码网络，这个网络包含了所有可能的搜索空间。

- 实验表明，用WFST构建的语音识别系统具有识别速度快，识别效果好的特性。

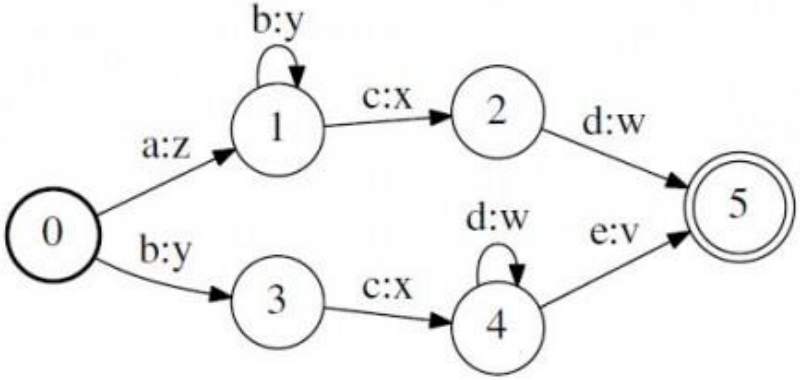
FSA=>WFST



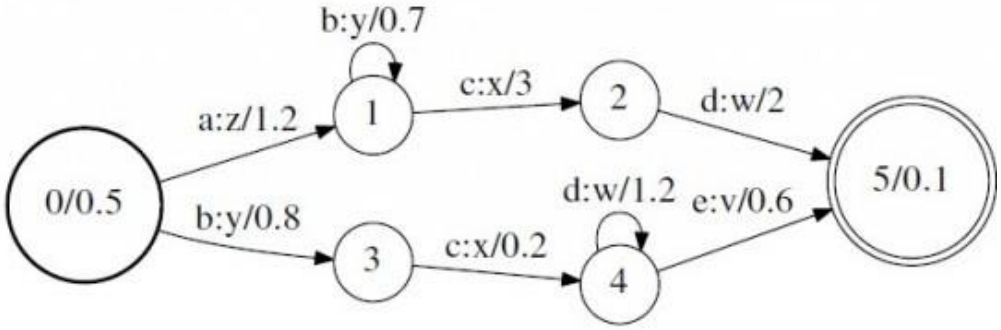
(a) Finite-State Acceptor (FSA)



(c) Weighted Finite-State Acceptor (WFSA)



(b) Finite-State Transducer (FST)



(d) Weighted Finite-State Transducer (WFST)

WFST半环及操作

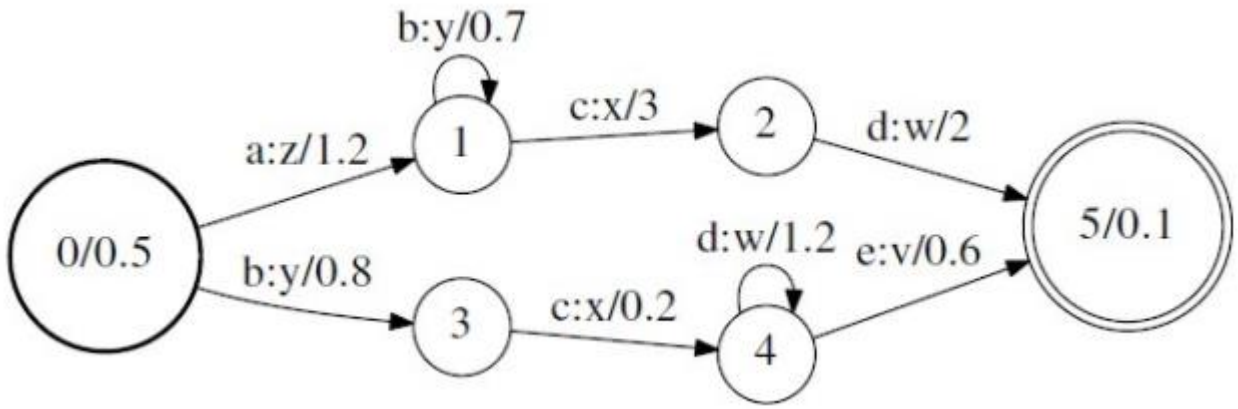
半环	集合	$\oplus$	$\otimes$	‘0’	‘1’
Boolean	{0,1}	$\vee$	$\wedge$	0	1
Probability	$\mathbb{R}_+$	$+$	$\times$	0	1
Log	$\mathbb{R} \cup \{-\infty, +\infty\}$	$\oplus_{\log}$	$+$	$+\infty$	0
Tropical	$\mathbb{R} \cup \{-\infty, +\infty\}$	min	$+$	$+\infty$	0

其中，$\oplus_{\log}$定义为$x \oplus_{\log} y = -\log(e^{-x} + e^{-y})$

WFST八元组

(1) Σ是一组有限的输入标签;
(2) $\wedge$是一组有限的输出标签;
(3) Q 是一组有限的状态;
(4) $I \subseteq Q$是一组初始状态;
(5) $F \subseteq Q$是一组终止状态;
(6) $E \subseteq Q \times (\Sigma \cup \{\in\}) \times K \times Q$是多组有限的转移;
(7) $\lambda: I \to K$是权重初始函数;
(8) $\rho: F \to K$是权重终止函数;

WFST八元组

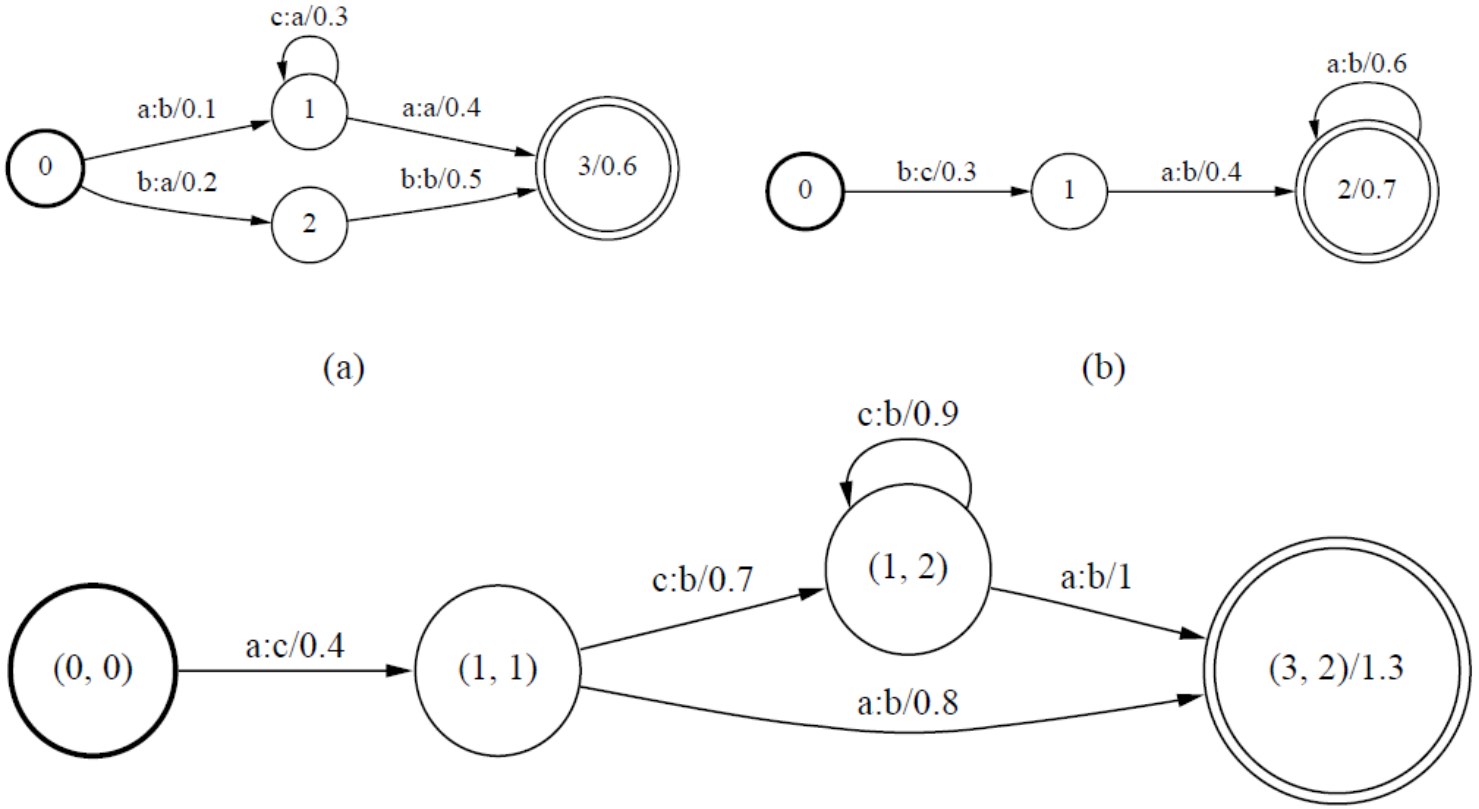


(1) $\Sigma = \{a, b, c, d, e\}$
(2) $\wedge = \{v, x, y, w, z\}$
(3) $Q = \{0,1,2,3,4,5\}$
(4) $I = \{0\}$
(5) $F = \{5\}$
(6) $E = \{(0, a, z, 1.2,1), (0, b, y, 0.8,3), (1, b, y, 0.7,1), (1, c, x, 3,2),$
$(2, d, w, 2,5), (3, c, x, 0.2,4), (4, d, w, 1.2,4), (4, e, v, 0.6,5)\}$
(7) $\lambda\,(0) = 0.5$
(8) $\rho(5) = 0.1$

WFST三大算法

- Composition
- Determinization
- Minimization

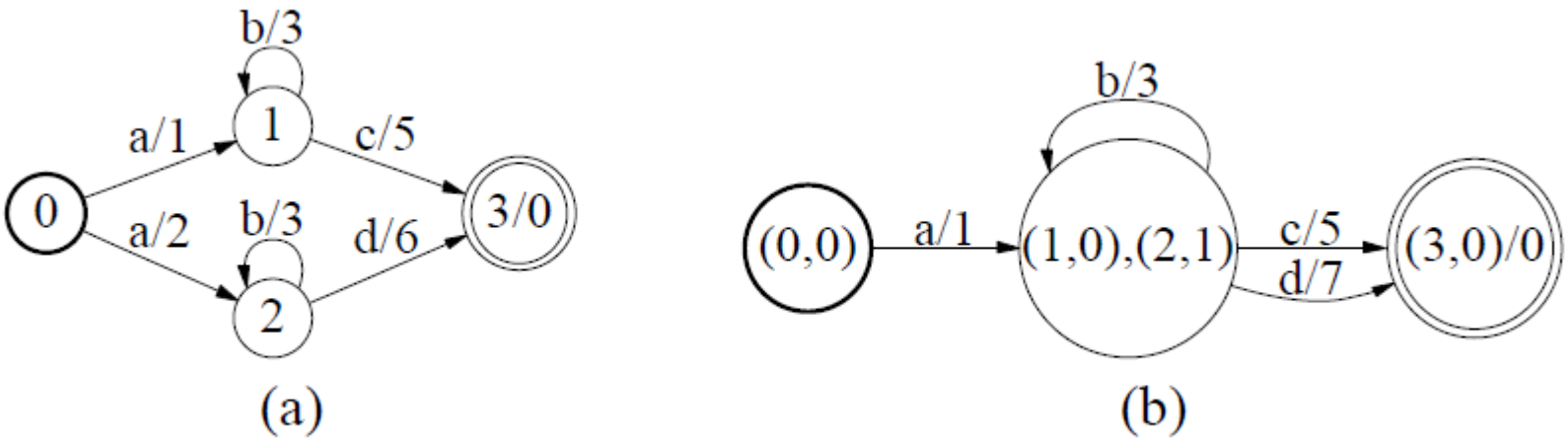
WFST的Composition



From: "Speech Recognition with Weighted Finite-State Transducers" by Mohri, Pereira and Riley (in Springer Handbook on Speech Processing and Speech Communication, 2008)

- 图(a)和图(b)Composition后生成了图(c)
- 第一个WFST的某个转移上的输出标签等于第二个WFST的某个转移上的输入标签，然后把这些转移上的label和weight分别进行操作。

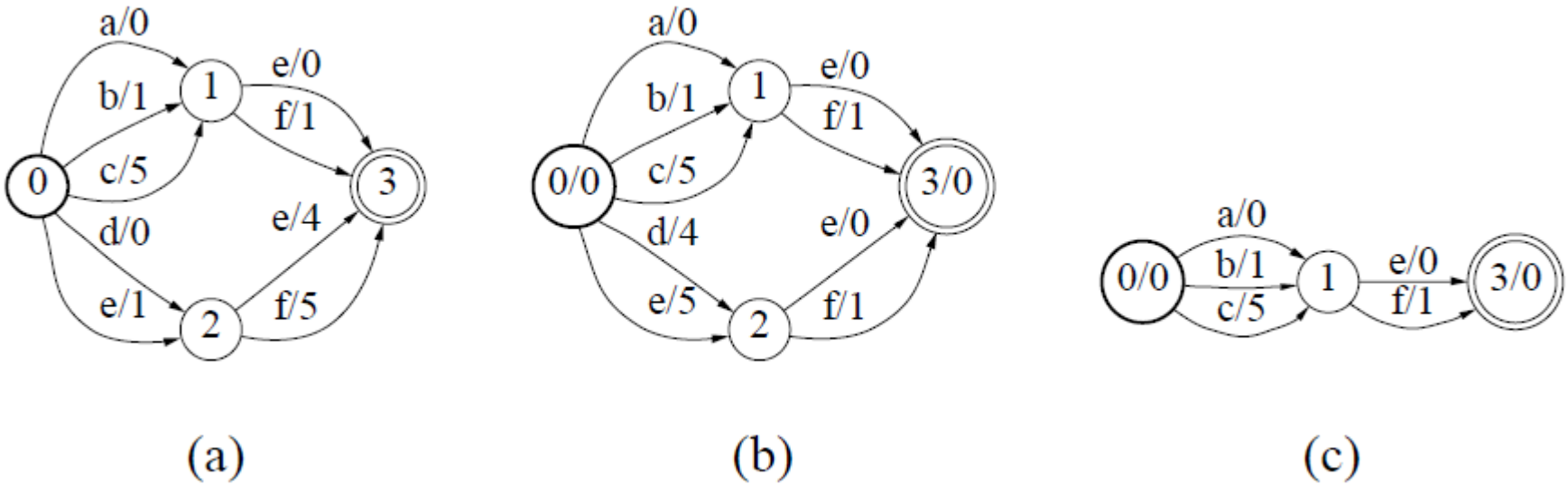
WFST的Determinization



From: "Speech Recognition with Weighted Finite-State Transducers" by Mohri, Pereira and Riley (in Springer Handbook on Speech Processing and Speech Communication, 2008)

- 图(a)Determinization后生成了图(b)

WFST的Minimization



From: "Speech Recognition with Weighted Finite-State Transducers" by Mohri, Pereira and Riley (in Springer Handbook on Speech Processing and Speech Communication, 2008)

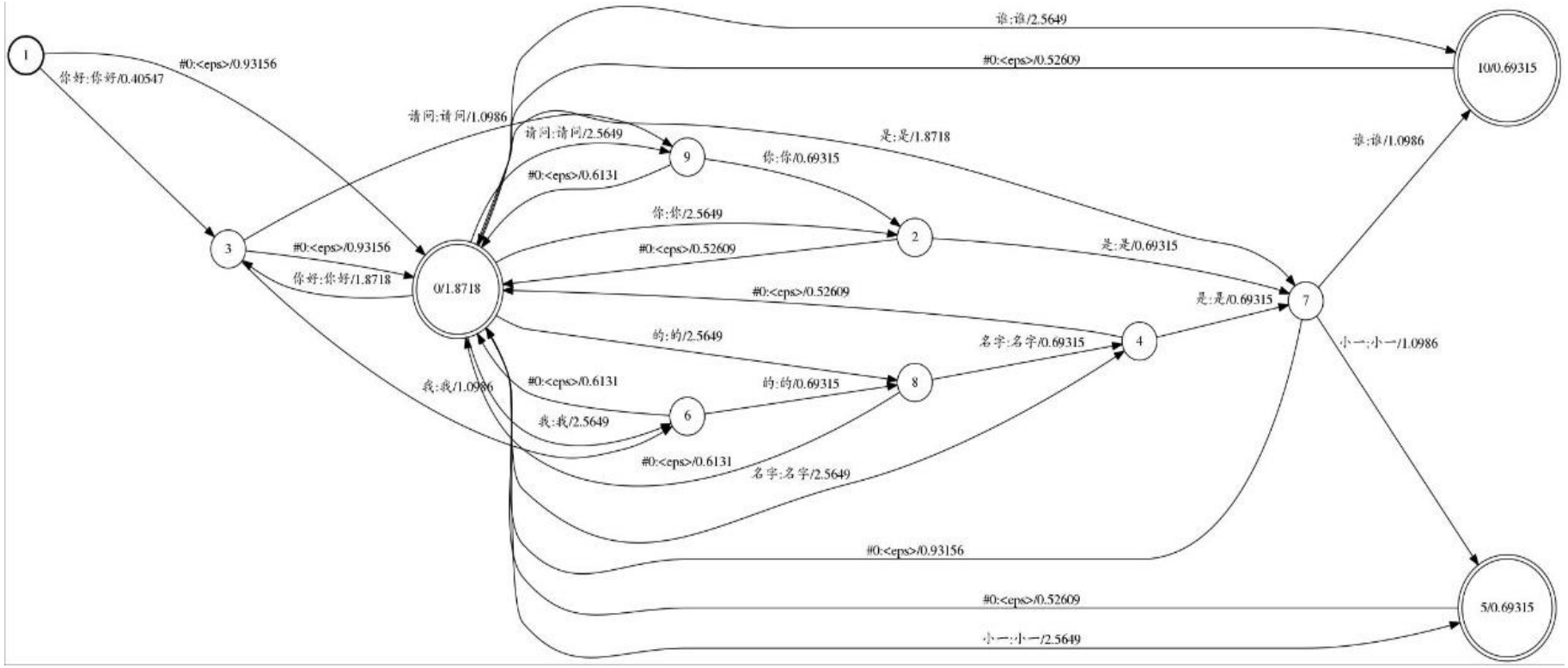
- 一般在做minimization前会先做一个权重推移（Weight Pushing），从图(a)变到图(b)
- 图(b)做Minimization后变为图(c)。

HCLG

- 大词汇量连续语音识别(LVCSR)所常用的四类模型：HMM、跨词三音子模型、词典以及语言模型，实际上是在不同粒度上描述了可能的搜索空间。

- 用H、C、L、G分别表示上述HMM模型、三音子模型、字典和语言模型的WFST形式。

语言模型G.fst


1
你好:你好/0.40547
#0:<eps>/0.93156
3
请问:请问/1.0986
请问:请问/2.5649
9
#0:<eps>/0.6131
你:你/0.69315
是:是/1.8718
谁:谁/2.5649
#0:<eps>/0.52609
10/0.69315
谁:谁/1.0986
你:你/2.5649
2
#0:<eps>/0.52609
是:是/0.69315
#0:<eps>/0.93156
你好:你好/1.8718
0/1.8718
#0:<eps>/0.52609
是:是/0.69315
7
的:的/2.5649
名字:名字/0.69315
4
小一:小一/1.0986
我:我/1.0986
#0:<eps>/0.6131
的:的/0.69315
8
6
我:我/2.5649
#0:<eps>/0.6131
名字:名字/2.5649
#0:<eps>/0.93156
#0:<eps>/0.52609
5/0.69315
小一:小一/2.5649

HCLG

H：包含HMM的定义，输入是转移id，转移id是表示状态的编码pdf_id，输出是上下文相关的三音子。

C：表示上下文关系，输入是三音子，输出是单音子。

L：发音词典，输入是单音子，输出是词。

G：用来编码语言模型的接收器，输入输出是一样的。

HCLG = min(det(H o C o L o G))

上面的o表示组合，det表示确定化，min表示最小化。

基于HCLG，输入是上下文依赖的HMM状态，输出是词序列。

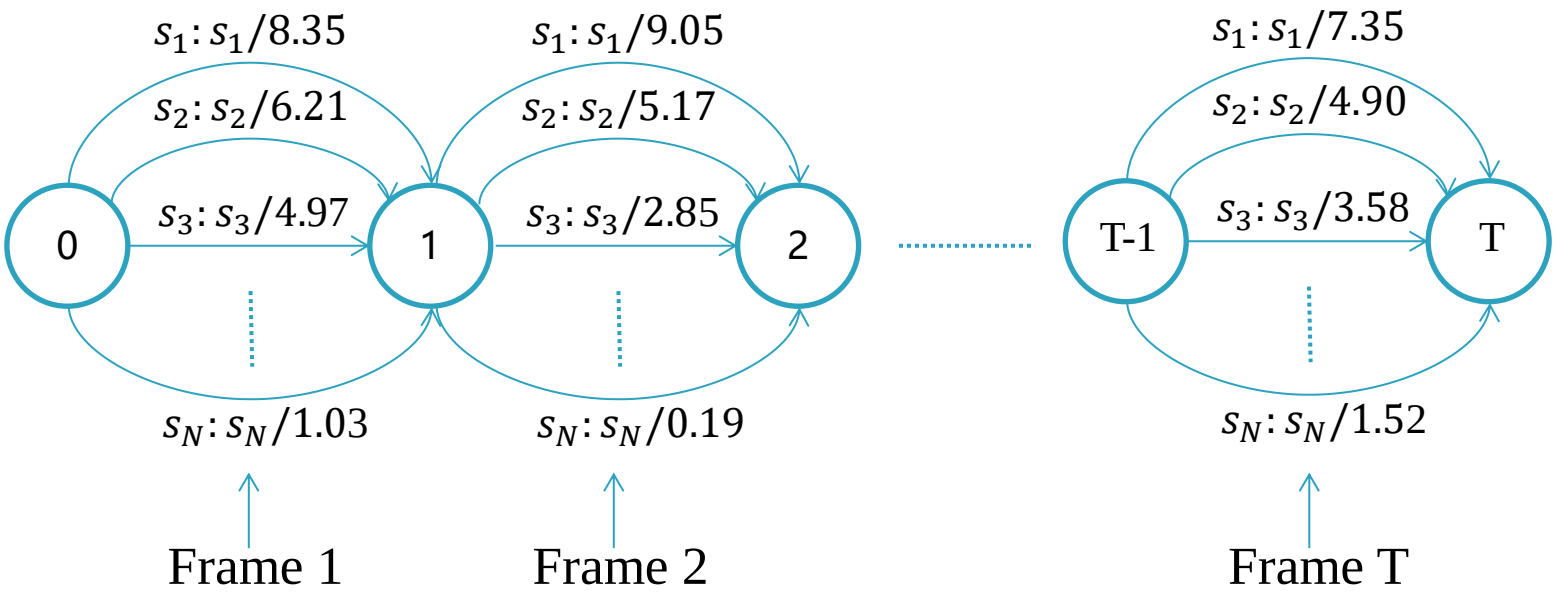
WFST解码

- 对T帧的语音进行解码，找到最相似的词序列和状态级的对齐序列。

- 定义U为一个拥有T+1个状态的WFSA，每两个状态之间的弧都带了时间、pdf-id信息。

WFST解码

- Acceptor U: T帧语音，N个状态



- Decoding:
  - $arg\max_{w}(U \circ HCLG)$

最优路径搜索

- 从搜索图S中，找出一个或多个最优的词序列。这在本质上属于搜索算法或解码算法的范畴。

- 全搜索几乎是不可能的。因此常采用基于一定裁剪路径的算法。

- 裁剪路径即放弃不可能的，或者说得分低的路径。例如：当该路径与最优路径得分的差值大于一定门限时，可以放弃该路径。

最优路径搜索

1、Viterbi Beam搜索算法

- 初始化
  初始化活动路径（最高层）
- 递推

For t=1 到 T
For 每一层次（指各个层次的语言和声学模型）
For HMM的每个活动状态
把每个活动路径向后扩展一帧至所有可以到达的状态
执行Viterbi计算
裁剪路径
End {活动状态}
End {每一层次}

Viterbi Beam算法是一个次优算法，最优路径有可能在开始时因得分过低而被裁剪掉。不过，在语音识别中次优算法也往往可以应用。

最优路径搜索

2、基于Viterbi的N-best算法

采用Viterbi搜索，在每个词层保留N个最优的前接路径，并分别向后扩展，在新的词层进行裁减后仍保留N个最优。

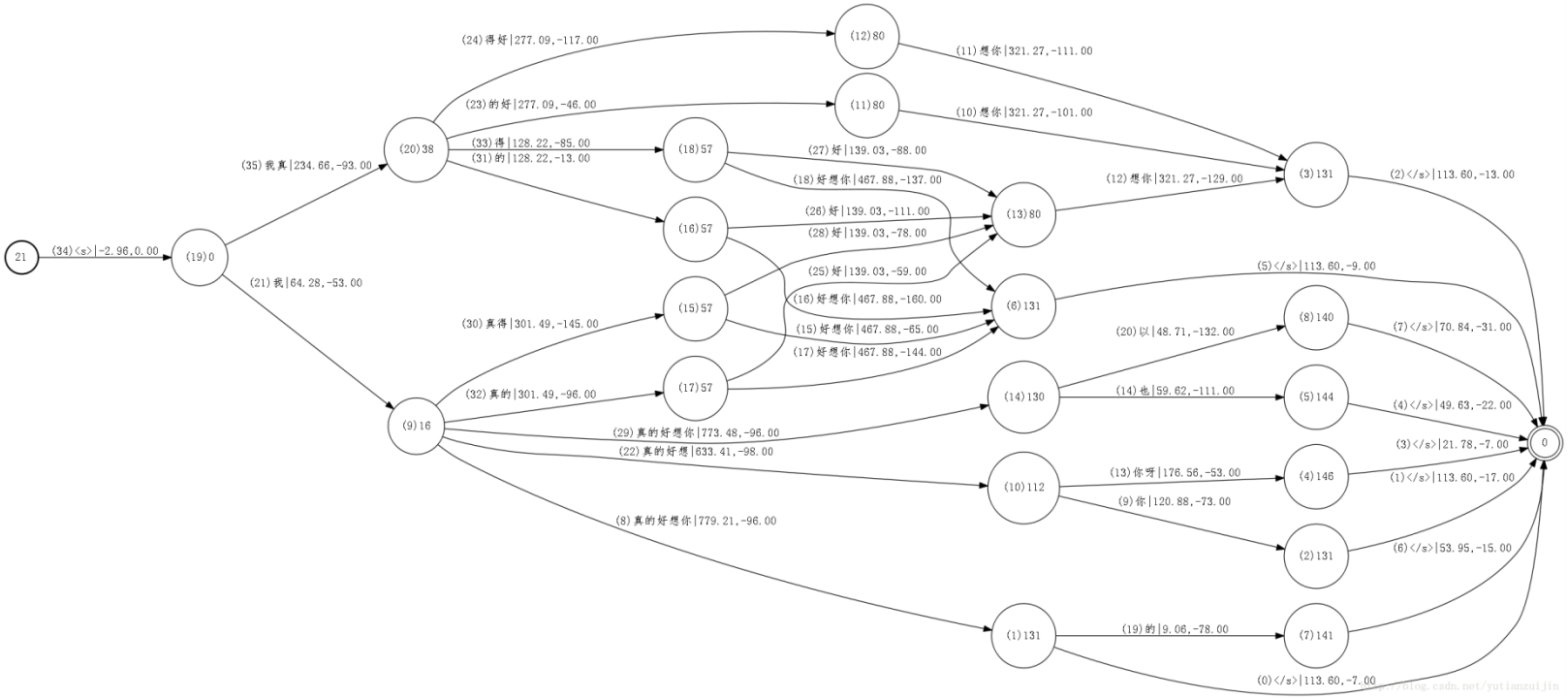
全部搜索结束后，选出最优的N个结果，并分别逐次回溯出N条路径。

词图Lattice

- 在实际的语音识别系统中，最优路径不一定与实际字序列匹配。
- 用来保存得分最靠前的多条候选路径，即N-best。

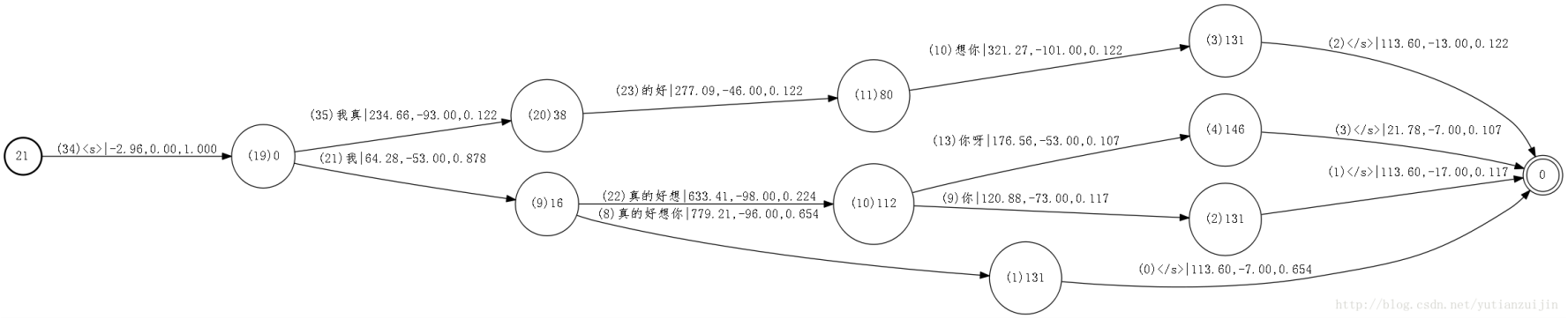
Lattice结构

- FST的形式，weight包括两部分（graph cost和acoustic cost），输入是transition-ids，输出是words。其中weight的graph cost包含LM+transition+pronunciation三部分。



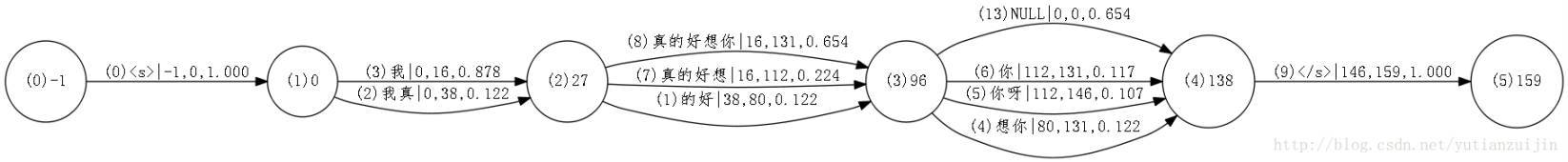
Lattice剪枝

- 通过限制N的大小，可以使得算法更加快，代价就是损失可能在Lattice生成束里的词序列。



混淆网络

- 混淆网络是一种特殊的Lattice，由原始Lattice经过变换生成。它要求Lattice中的每条路径都必须经过所有的节点。



Thank you!

Any questions?