

OMax etc.

OMax Bros (G rard Assayag, Georges Bloch, Marc Chemillier, Shlomo Dubnov)

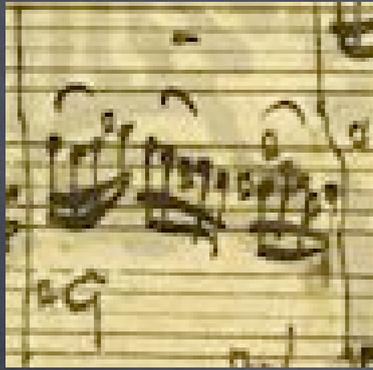
Sommario

- Il principio della re-iniezione stilistica
- i sistemi *off-line*
- Il *factor oracle* (oracolo di fattori)
 - modello d'apprendimento
 - modello per ricreare (generare) nuove sequenze
- struttura di OMax
- I tre mondi : simbolico, audio e spettrale
- Il futuro...
- esempi

Apprendimento statistico / modelli di sequenze musicali

- I modelli delle sequenze prendono la loro ispirazione dalle teorie dei linguaggi formali / automi
- così si catturano elementi importanti dello stile
- La complessità multidimensionale della musica richiede una rappresentazione intelligente
- la sequenza è appresa a poco a poco in un modello associato ad un schema probabilistico di predizione
- è un apprendimento "agnostico" (senza conoscenza a priori dello stile) e fatto senza supervisione dell'utilizzatore
- la classificazione si fa quando si suona
- la generazione utilizza uno schema di similitudine contestuale

Off-line Style Modeling



Jaco Pastorius

Clone



Bernard Lubat

Original



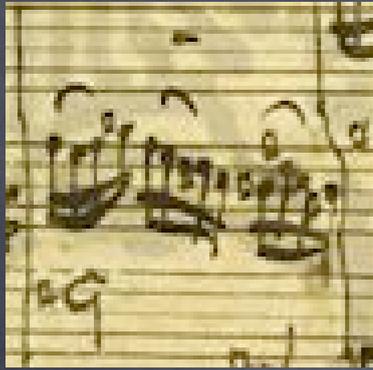
Clone



Passare gli esempi



Off-line Style Modeling



Jaco Pastorius

Clone



Bernard Lubat

Original



Clone



Passer les exemples



Off-line Style Modeling



Jaco Pastorius

Clone



Bernard Lubat

Original



Clone



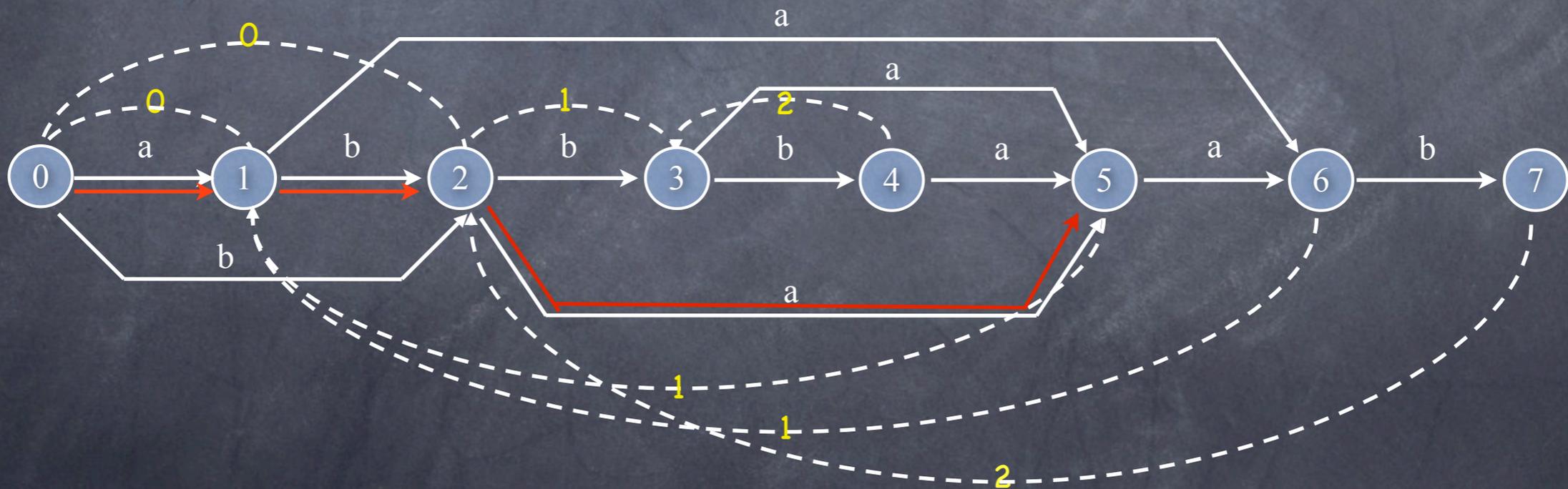
Passer les exemples



Struttura dell'oracolo

1 oracolo $\left\{ \begin{array}{l} \uparrow \text{una sequenza} \\ \text{2 serie di frecce} \\ \downarrow \text{una classificazione} \end{array} \right.$

abbbaab



Si vede che si trova a-b-a che non è parte della sequenza originale

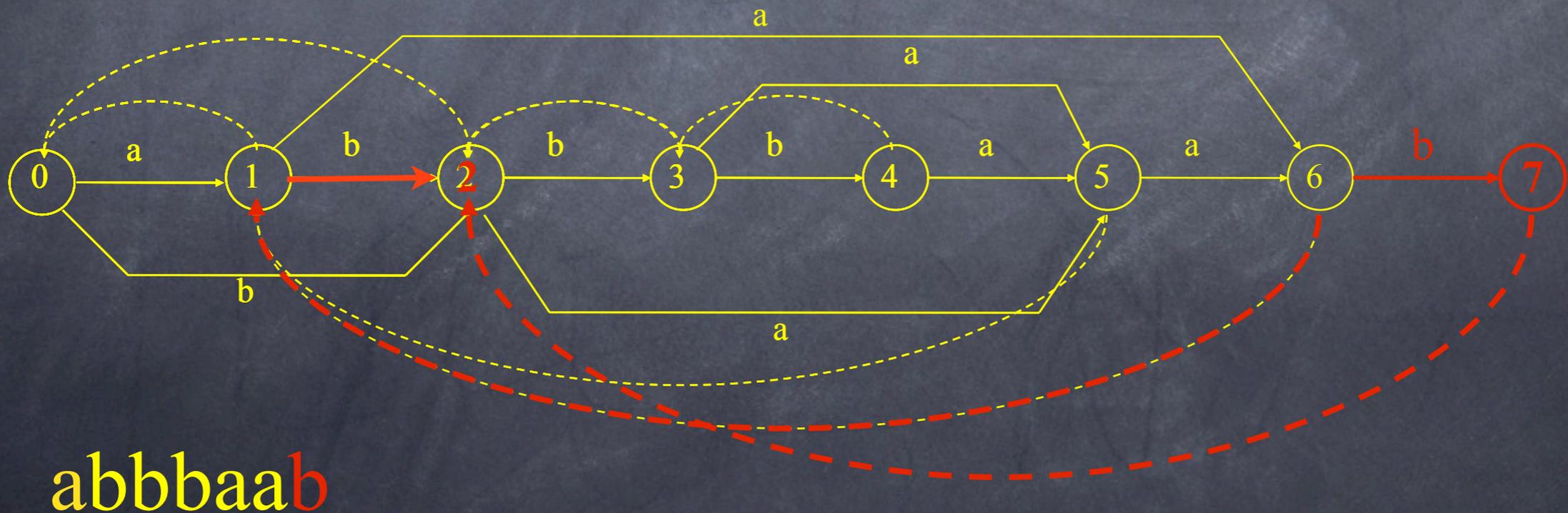
Struttura dell'oracolo

algoritmo add_letter:
Allauzen & al. SOFEN99

1. creazione dello stato 7
2. creazione della transizione chiamata b
3. si salta alla fine del suffisso uscendo dallo stato 6 (verso stato 1); dunque $k=1$
4. ciclo **While** : c'è una transizione di tipo b dallo stato 1 (in direzione di stato 2), le condizioni dello loop non sono compiute. Si esce subito dallo ciclo
8. **If** $k = -1$. Non è il caso, perchè $k=1$
9. **Else** si segue la transizione, verso stato 2
10. ...e si fa un legame suffisso dello stato 7 al stato 2
11. **Return** è finito !

```
Function add_letter(Oracle( $p = p_1 p_2 \dots p_m$ ),  $\sigma$ )  
1. Create a new state  $m + 1$   
2. Create a new transition from  $m$  to  $m + 1$  labeled by  $\sigma$   
3.  $k \leftarrow S_p(m)$   
4. While  $k > -1$  and there is no transition from  $k$  by  $\sigma$  Do  
5. Create a new transition from  $k$  to  $m + 1$  by  $\sigma$   
6.  $k \leftarrow S_p(k)$   
7. End While  
8. If ( $k = -1$ ) Then  $s \leftarrow 0$   
9. Else  $s \leftarrow$  where leads the transition from  $k$  by  $\sigma$ .  
10.  $S_{p\sigma}(m + 1) \leftarrow s$   
11. Return Oracle( $p = p_1 p_2 \dots p_m \sigma$ )
```

Figure 4. Add a letter σ to Oracle($p = p_1 p_2 \dots p_m$) to get Oracle($p\sigma$)



Struttura dell'oracolo

Algorithme add_letter:
Allauzen & al. SOFEN99

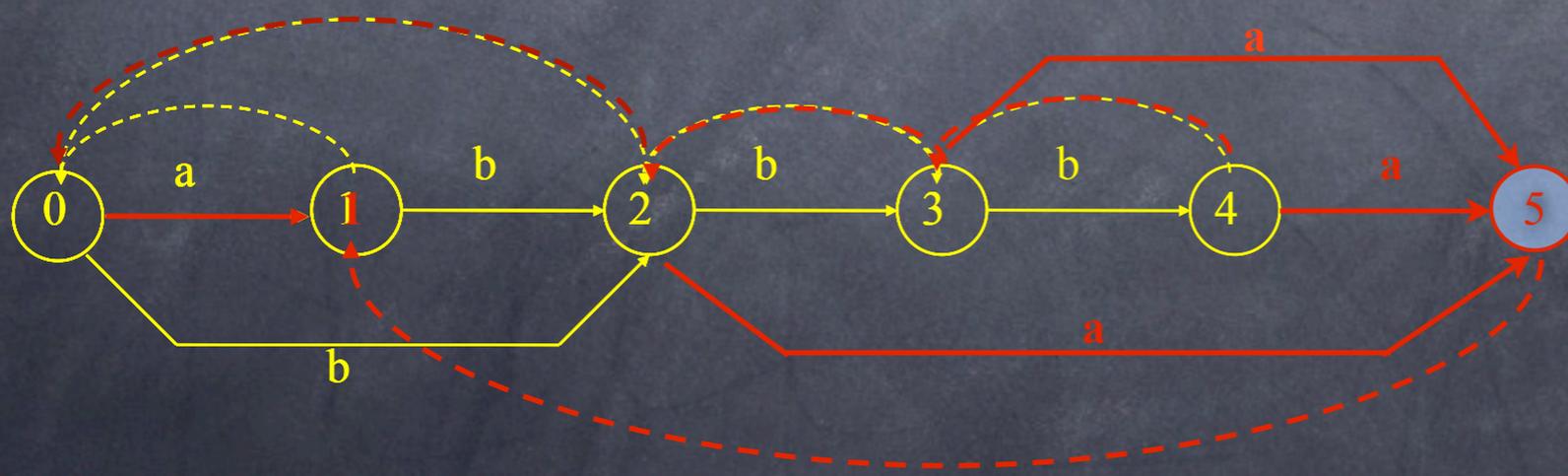
1. creazione dello stato 5
2. si fa una transizione chiamata a
3. si segue il suffisso che parte dallo stato 4 (fino al stato 3). Dunque $k=3$
4. ciclo **While** : $k \neq -1$ E NON c'è di transizione chiamata a
5. si fa un legame posteriore stato 3 \rightarrow stato 5 chiam. a
6. si segue il suffisso dallo stato 3 fino al stato 2 ($k=2$)
- 4bis. $k \neq 1$ E NON c'è di transizione chiamata a
- 5bis. si fa un legame stato 3 \rightarrow stato 5 chiamato a
- 6bis. si segue il suffisso da stato 2: si arriva al stato 0
- 4ter. c'è un legame stato 0 \rightarrow stato 1 del tipo a
7. per conseguenza si esce del ciclo
8. **If** $k = -1$. $k=0$, non è il caso
9. **Else** si segue la transizione (al stato 1)
10. ...e si fa un suffisso dal stato 5 al stato 1

```

Fonction add_letter(Oracle( $p = p_1 p_2 \dots p_m$ ),  $\sigma$ )
1. Create a new state  $m + 1$ 
2. Create a new transition from  $m$  to  $m + 1$  labeled by  $\sigma$ 
3.  $k \leftarrow S_p(m)$ 
4. While  $k > -1$  and there is no transition from  $k$  by  $\sigma$  Do
5.     Create a new transition from  $k$  to  $m + 1$  by  $\sigma$ 
6.      $k \leftarrow S_p(k)$ 
7. End While
8. If ( $k = -1$ ) Then  $s \leftarrow 0$ 
9. Else  $s \leftarrow$  where leads the transition from  $k$  by  $\sigma$ .
10.  $S_{p\sigma}(m + 1) \leftarrow s$ 
11. Return Oracle( $p = p_1 p_2 \dots p_m \sigma$ )
    
```

Figure 4. Add a letter σ to Oracle($p = p_1 p_2 \dots p_m$) to get Oracle($p\sigma$)

11. Return è finito !



abbba

Un esempio più complesso :
quando arriva il "a" dello stato 5, si debbe trascorrere 3 legami suffissi prima di ritrovare un "a". Si costruisce due legami posteriori e, finalmente, il legame suffisso. Da notare: il suffisso va verso la prima apparizione del "a"

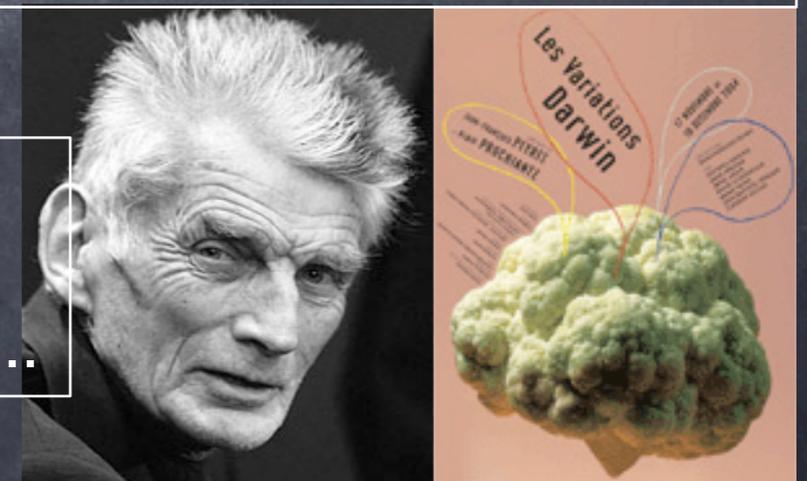
modelizzazione : re-combinazione

All known all white bare white body fixed [4] **one yard legs joined like sewn** [5]. Light heat white floor one square yard never seen. White walls one yard by two white ceiling one square yard never seen. Bare white body fixed only the eyes only just. Traces blurs light grey almost white on white. Hands hanging palms front white feet [6] **heels together right angle.** [2] **Light heat white planes shining white** [3] bare white body fixed ping fixed elsewhere. Traces blurs signs no meaning light grey almost white. Bare white body fixed white on white invisible. Only the eyes only just light blue almost white. Head haught eyes light blue almost white silence within. Brief murmurs only just almost never all known. Traces blurs signs no meaning light grey almost white. [5] **Legs joined like sewn heels together right angle.** [6] Traces alone white. Light heat white walls [3] **shining white one yard** [4] by two. Bare white body fixed one yard ping fixed elsewhere. Traces blurs signs no meaning light grey almost white. White feet toes joined like sewn heels together right angle invisible. Eyes alone uncover given blue light blue almost white. Murmur only just almost never one second perhaps not alone. **Given rose only just** [1] bare white body fixed one yard white on white invisible. All white all known murmurs [1] **only just almost never always the same all known.** **Light heat** [2] hands hanging palms front white on white invisible. Bare white body fixed ping fixed elsewhere. Only the eyes only just light blue almost white fixed front."

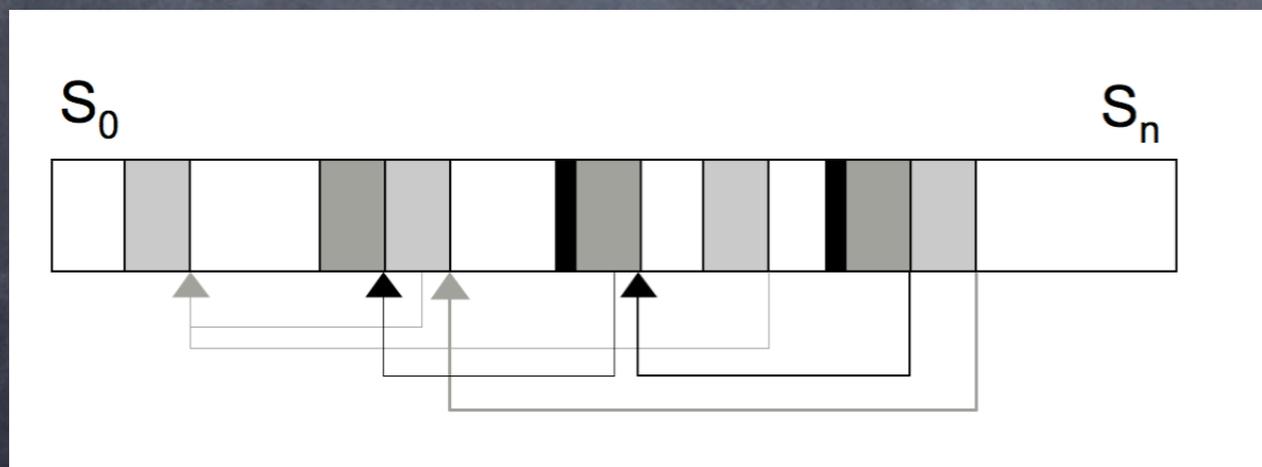
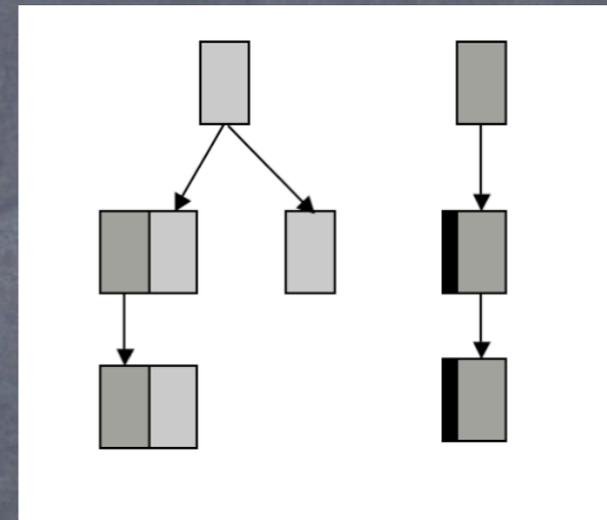
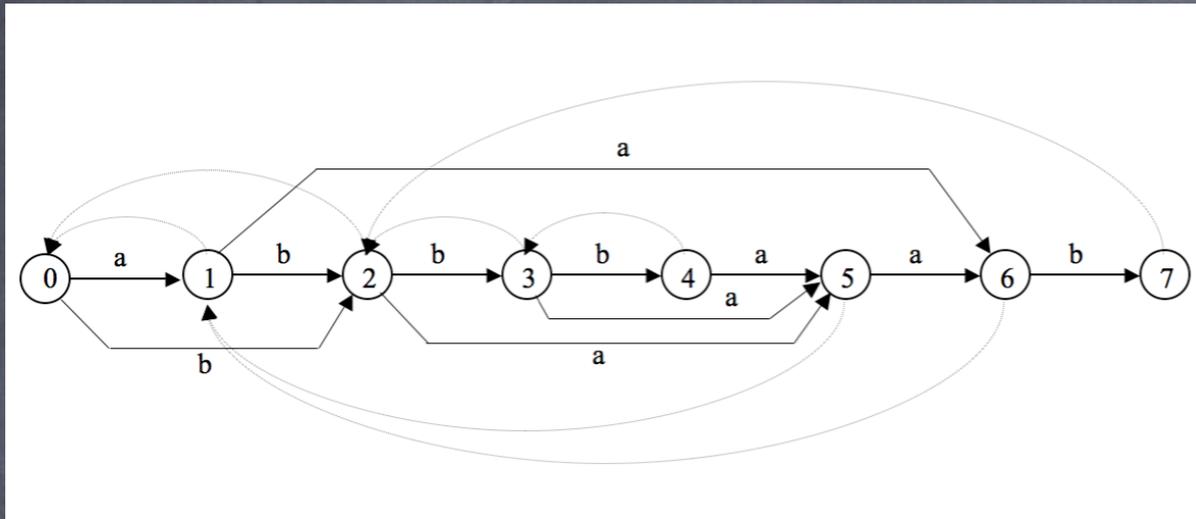
Samuel Beckett, *Ping* (1966-67)

Given rose only just almost never always the same all known. Light heat white planes shining white one yard legs joined like sewn heels together right angle. Light heat white...

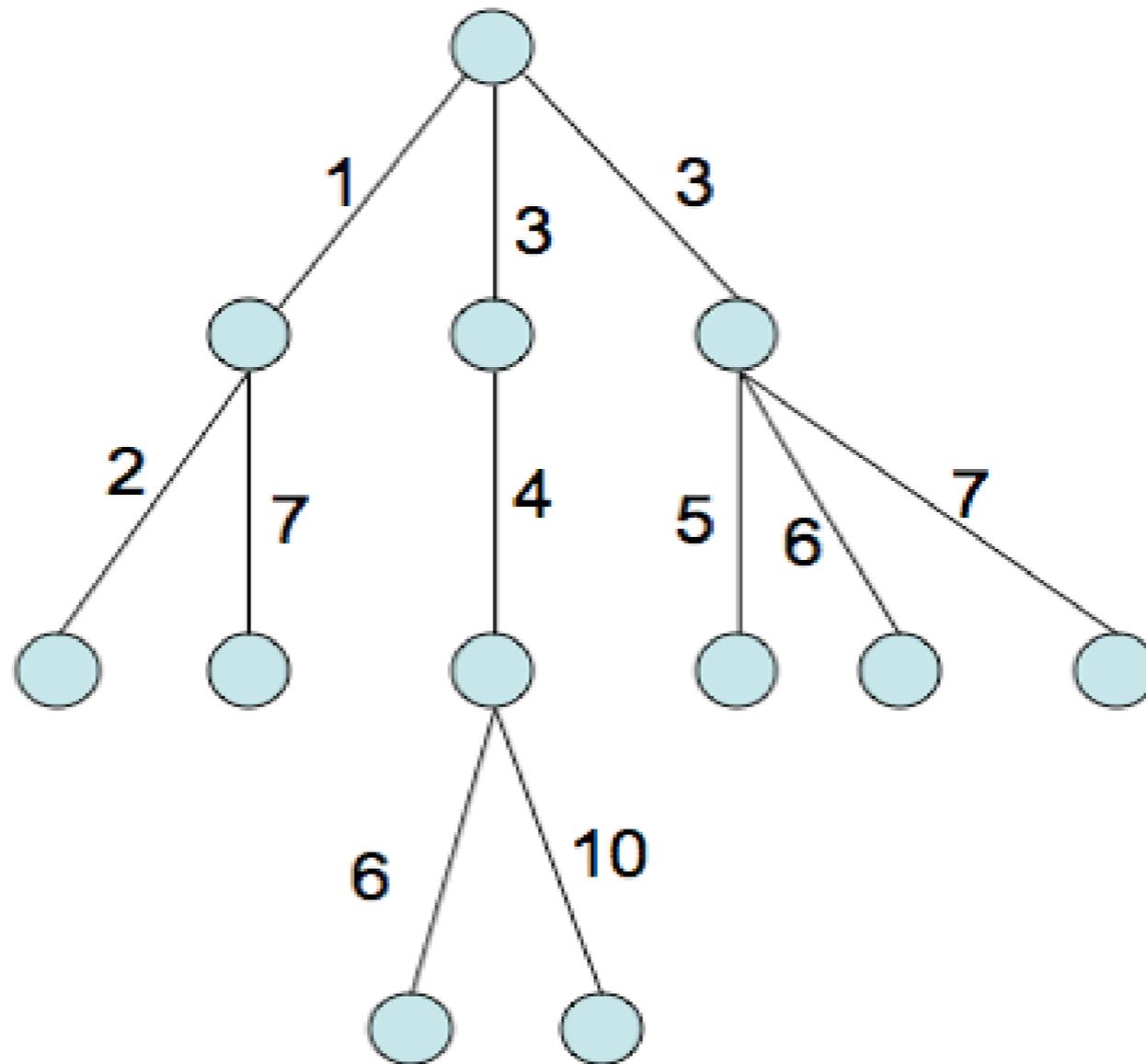
Testo generato in pseudo-Beckett



Alberi di legami suffissi

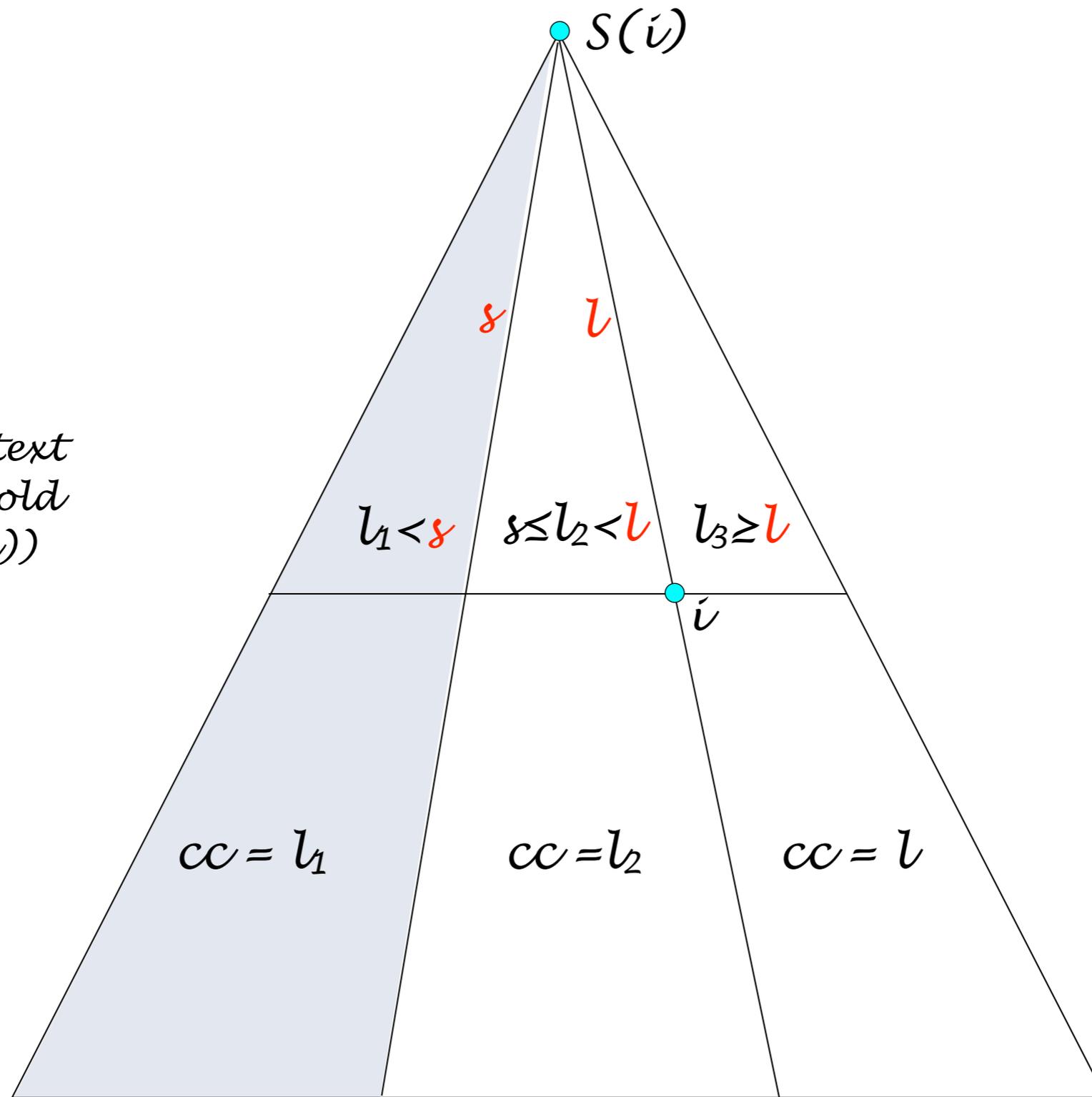


Alberi di legami suffissi

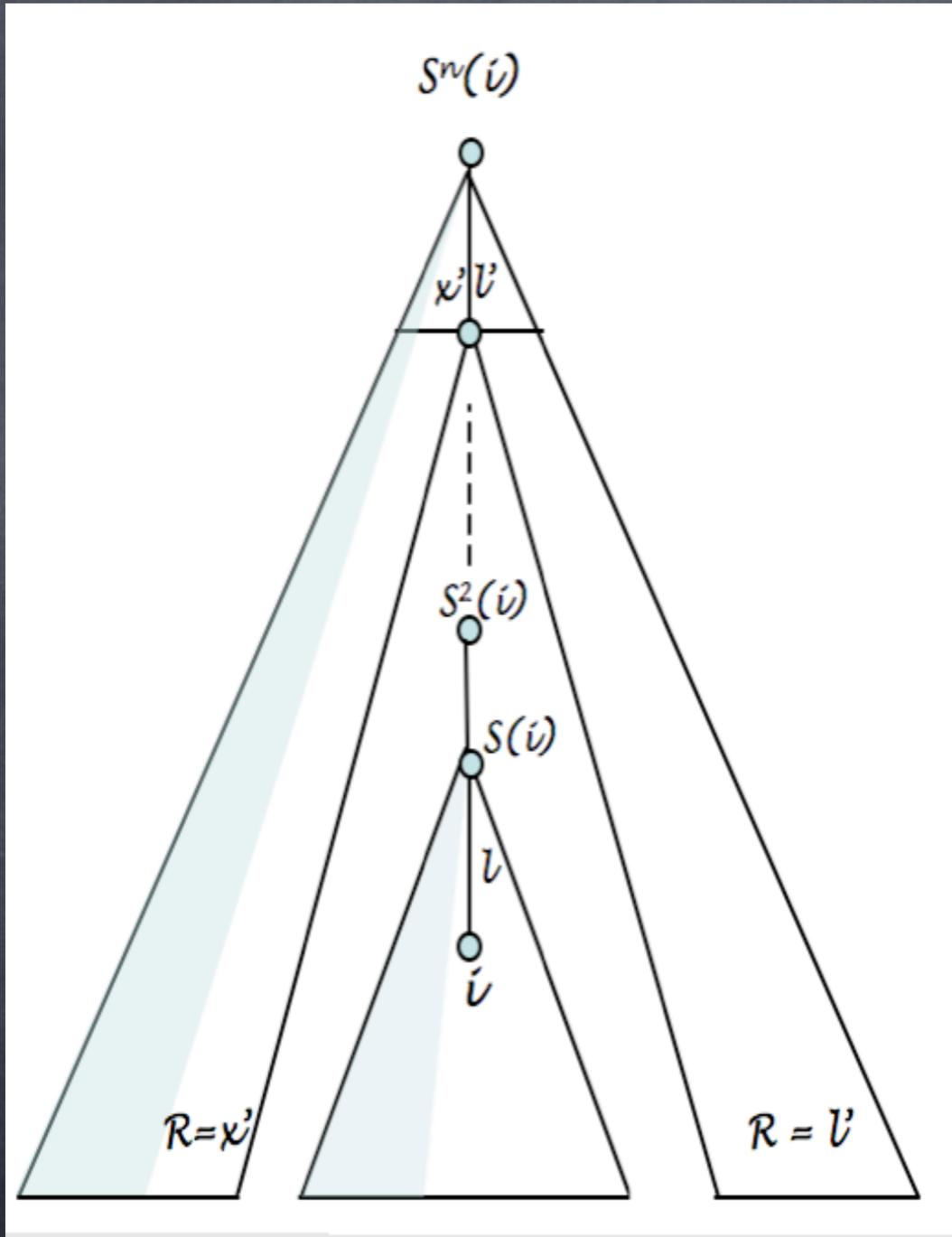


Alberi di legami suffissi

$cc = \text{common context}$
 $s = \text{context threshold}$
 $l = \text{length}(i, S(i))$



Alberi di legami suffissi: navigazione



Strategia di navigazione

Quando si vuole cambiare di contesto

- Dal punto corrente, si esplora il futuro
- Per ogni punto, esplorare l'albero et recuperare le soluzioni possibili
- ordinare in funzione della lunghezza del contesto
- Applicare una distribuzione probabilistica in favore del contesto et della qualità ritmica
- Saltare al seletto punto candidato
- Mettere il punto scelto in una lista di tabu (per alcuno tempo)

Max/MSP

Cattura dello strumento

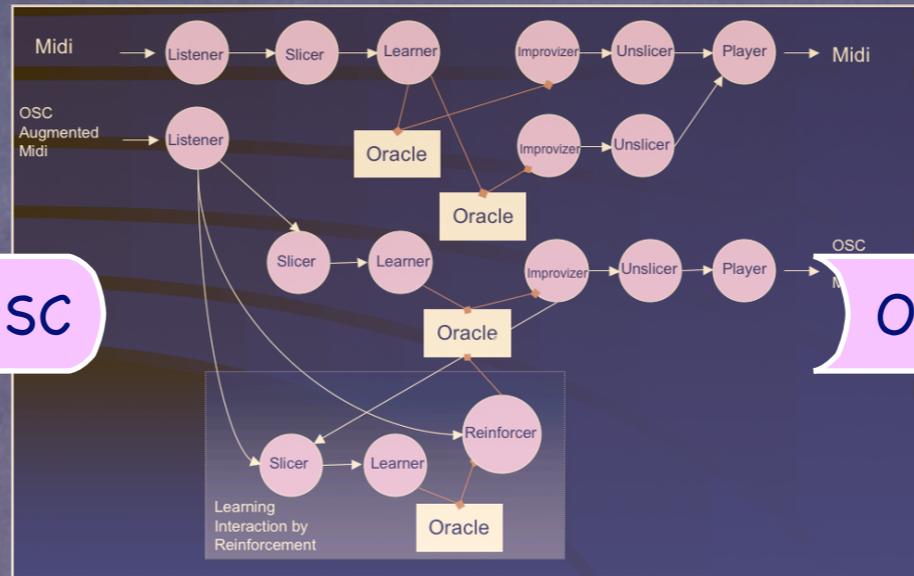
estrarre i descrittori del segnale

Estrarre gesti

...

Video

OpenMusic



OSC

OSC

Max/MSP

Rendizione del Segnale

sintesi concatenativa

correspondanza dei gesti

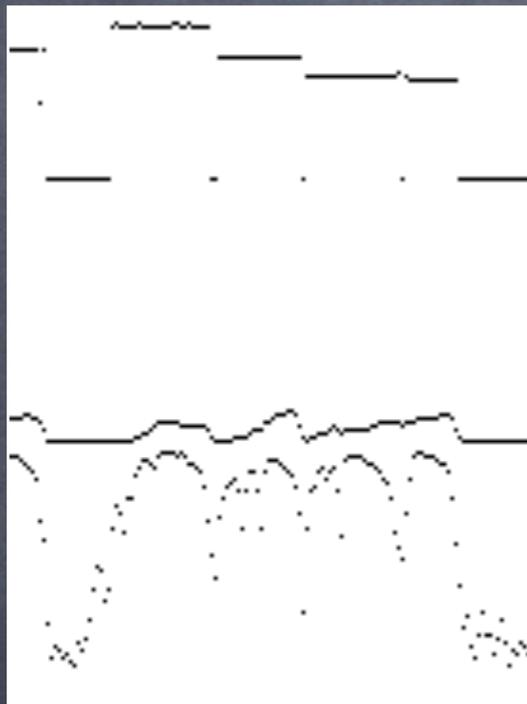
...

Video

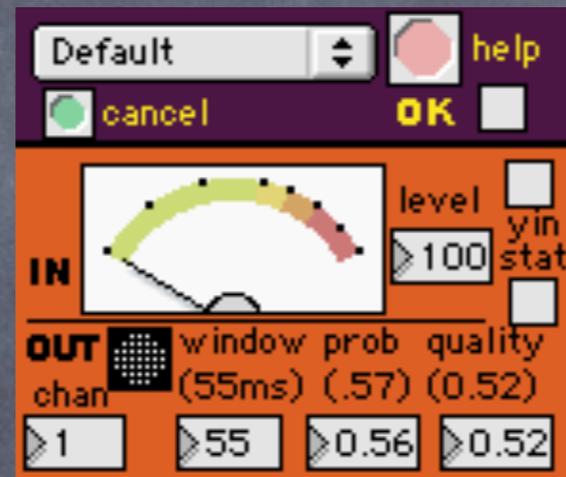
Sound and video information: Rappresentazione musicale aumentata

Detezione delle note: un problema

- yin~ Cheveigné/Kawahara
- yin_GB: statistiche sul yin~
- Transcribe : Arshia Cont



uscita del yin~



Yin_GB



Moog pianobar

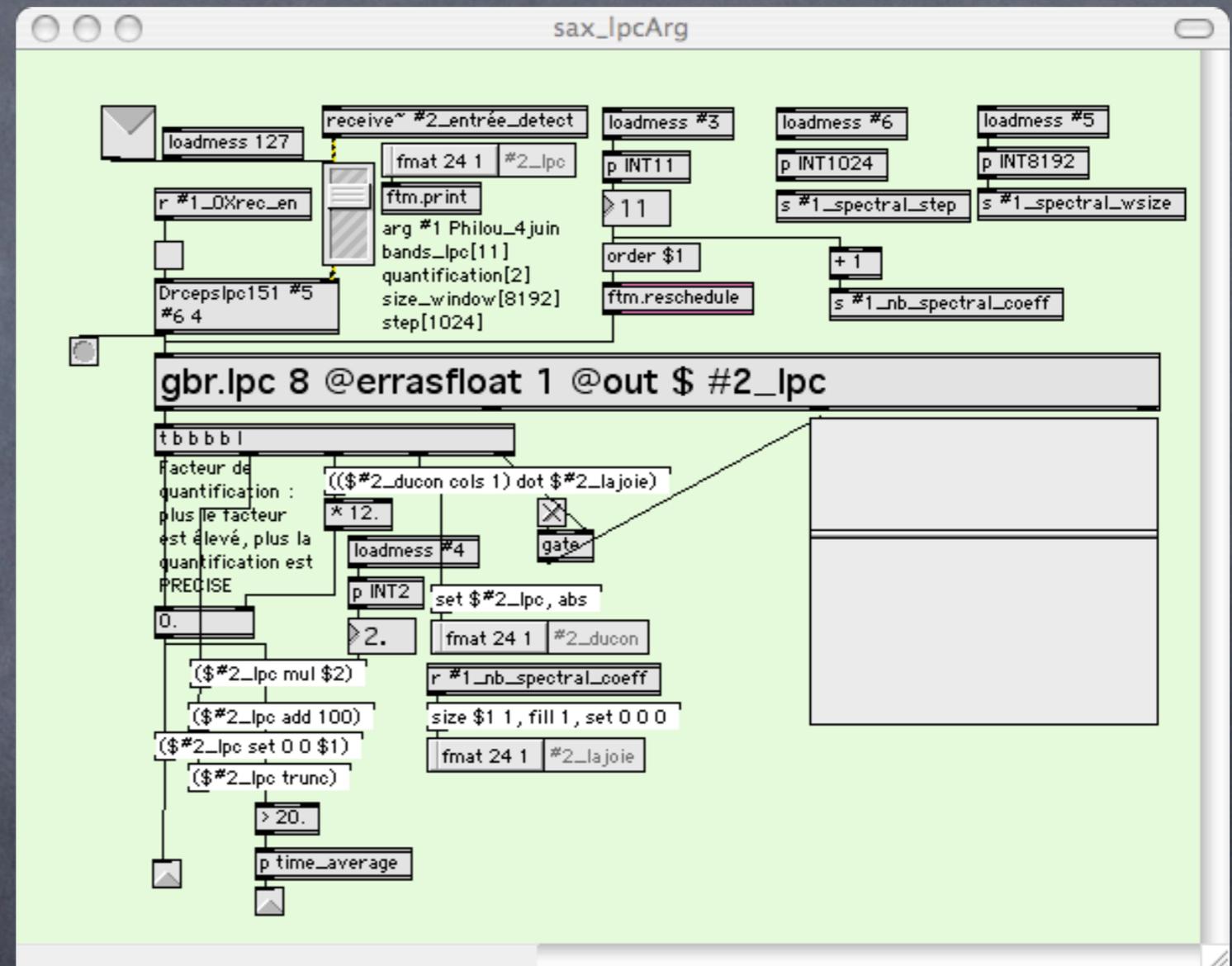
Cattura dei gesti: l'esempio del Moog pianobar

- naturalmente puo essere utilizzato per altri arti scenichi

Dimostrare yin_GB

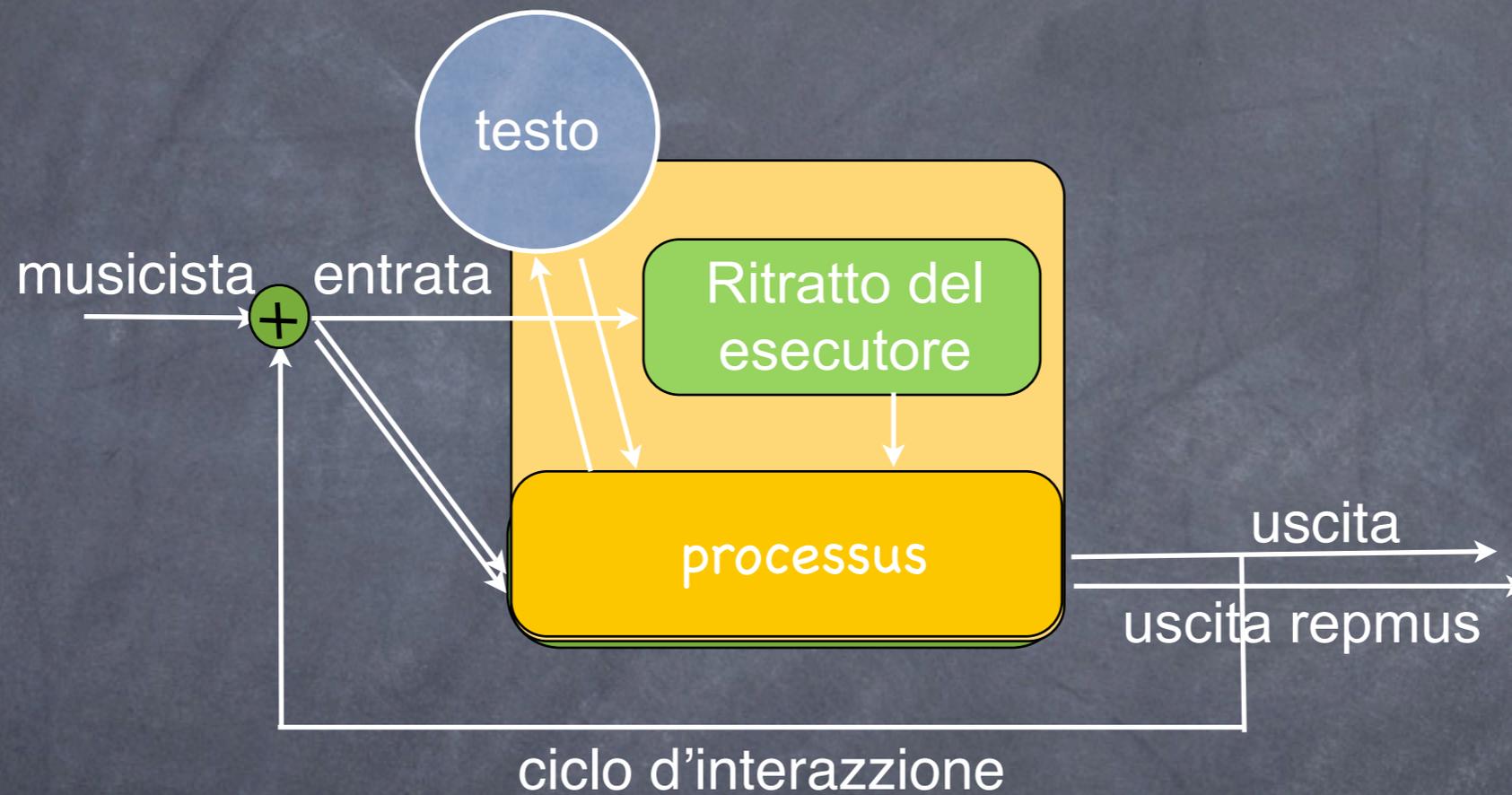
Entrata spettrale: classificazione dal sono proprio

- Presente: quantificazione selvaggia dei descrittori
- Perché una quantificazione ?
- Problema della distanza per costruire una relazione d'equivalenza
- Futuro: geometria di Bregman
- Divergenza di Kulback-Leibler
- Già usato nel "guidage"



Dimostrare spettrale
+ VIDEO ?
CAMBIARE il computer

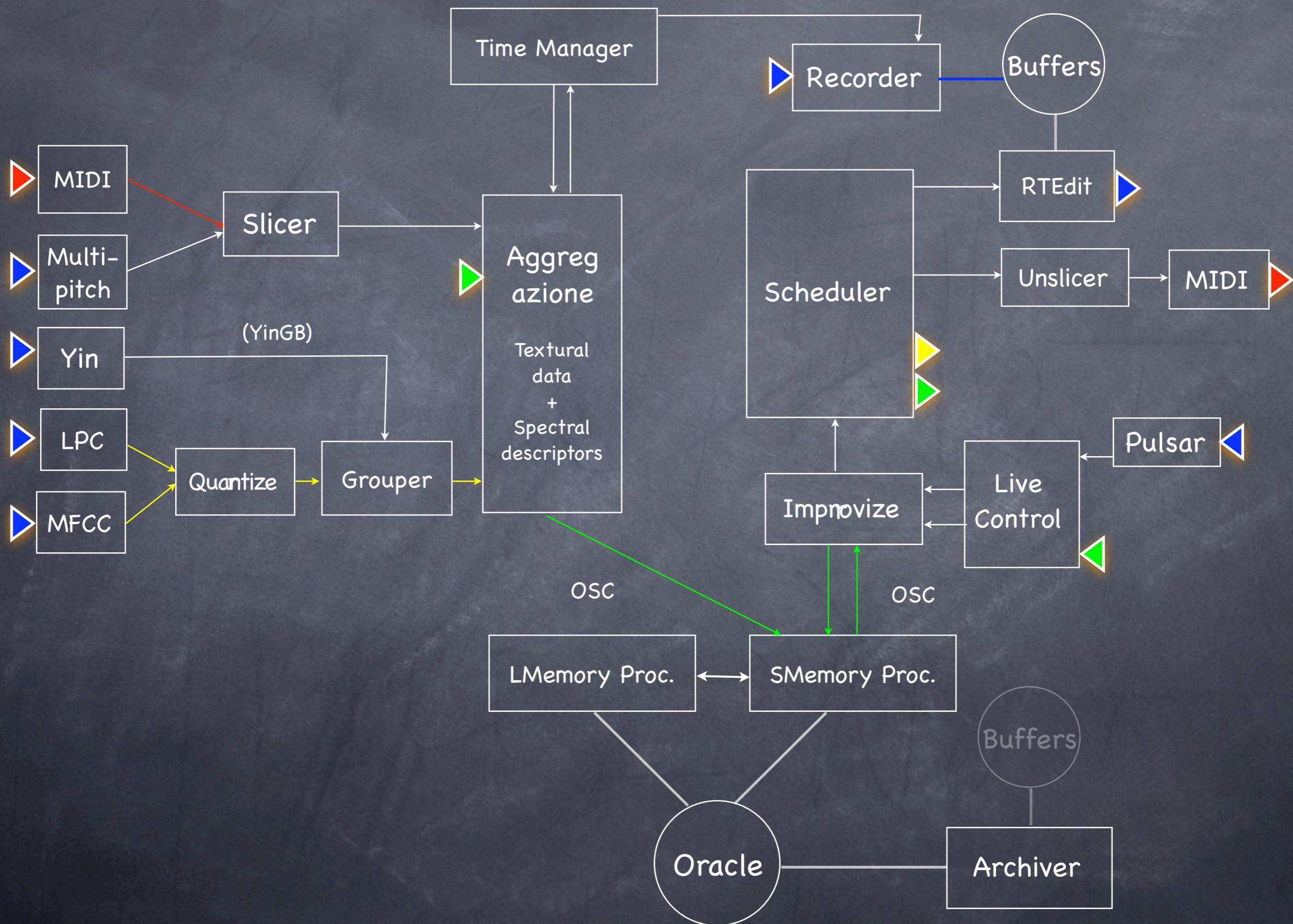
il ciclo d'interazione: una rivista



Cattura

Analisi strutturale

Rendizione



Bibliographie

<http://www.ircam.fr/equipes/repmus/OMax>

- Bloch, G., Dubnov, S., Assayag, G., “Introducing Spectral and Video Features into the OMax Improvisation System”, *Proc. ICMC’08*, Belfast, August 2008.
- Assayag, Gérard, Bloch, Georges, “Navigating the Oracle”, *Proc. ICMC’07*, Copenhagen, August 2007.
- Assayag, G., Bloch, G., Chemillier, Marc, “OMax-Ofon”, *Proc. SMC’06*, Marseille, June 2007.
- Assayag, G., Bloch, G., Chemillier, M., Cont, Arshia, Dubnov, Shlomo, “OMax Brothers : a Dynamic Topology of Agents for Improvization Learning”, *Proc. ACM Multimedia ‘06*, Santa Barbara, October 2006.
- Dubnov, S., Assayag, G., Lartillot, Olivier, Bejerano, G., "Using Machine-Learning Methods for Musical Style Modeling", *IEEE Computers*, 36 (10), pp. 73-80, Oct. 2003.
- de Cheveigné, Alain, Kawahara, H., “YIN, a fundamental frequency estimator for speech and music”, *J. Acoust. Sco. America*, 111, 1917-1930, 2002.
- Bloch, G., Chabot Xavier, Dannenberg, Roger, "A Workstation in Live Performance: Composed Improvization", *Proceedings of International Computer Music Conference*, The Hague, Netherlands, 1986.
- Chadabe, Joel, “Some reflections on the Nature of the Landscape within which Computer Music Systems are Designed”, in *Computer Music Journal*, 1, 3 (1977).