Statistical parametric speech synthesis

- Class slides

What we will cover in this class

- Brief recap of video content and Q&A
- Discussion points

Orientation

- Unit selection
  - selection of waveform units based on
    - target cost
    - join cost

e.g., the **IFF** formulation, which is based only on the **linguistic specification**

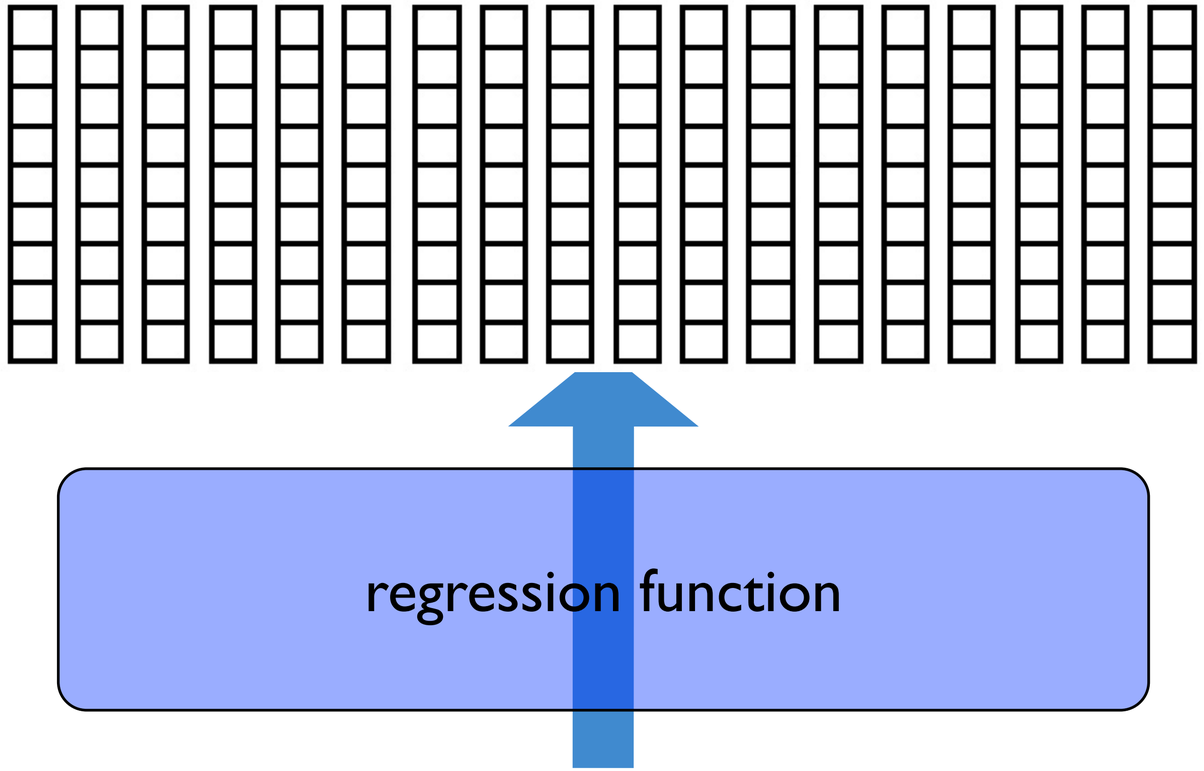
- Speech signal modelling
  - generalised source+filter model
- Statistical parametric synthesis
  - predict **speech parameters** from **linguistic specification**

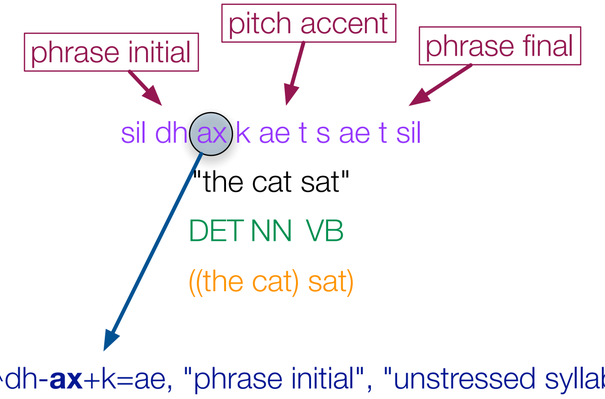
There are several ways to do this, but we need to be able to

- **separate** excitation & spectral envelope
- **reconstruct** the waveform

A **regression** task!

Orientation

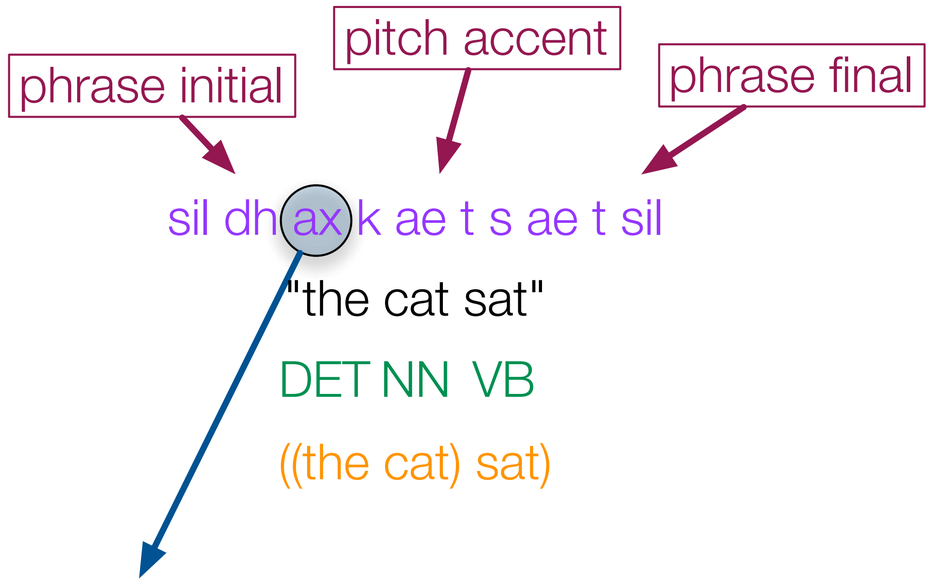




- Statistical parametric synthesis
  - predict **speech parameters** from **linguistic specification**

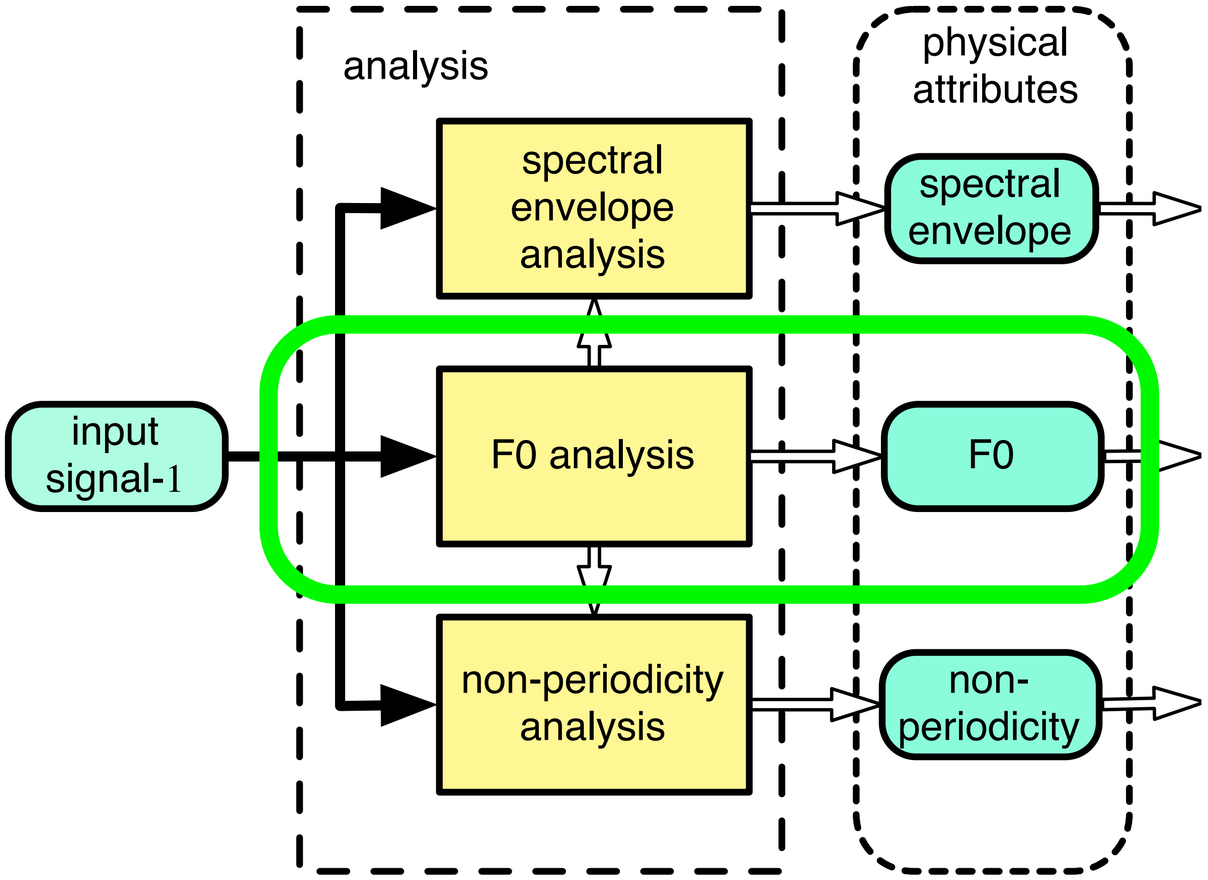
What are the input features ?

Just the linguistic features !



input feature vector

What are the output features (i.e., speech parameters) ?

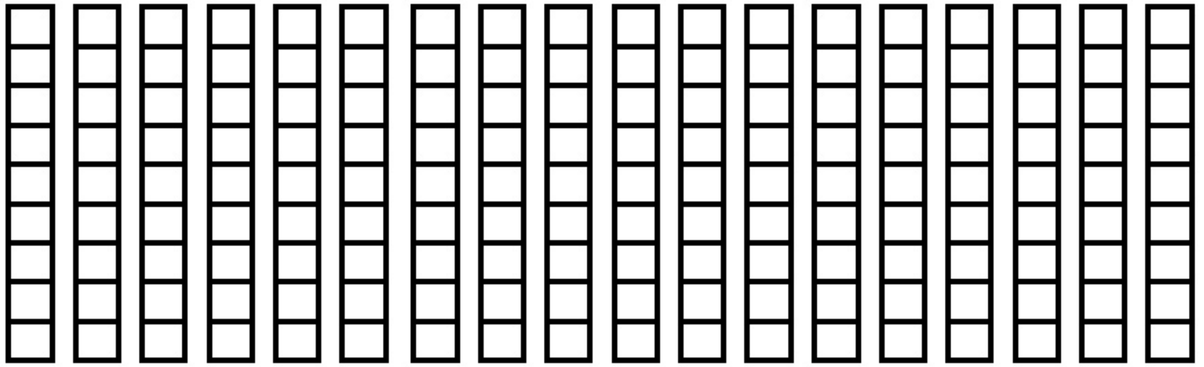


speech parameters

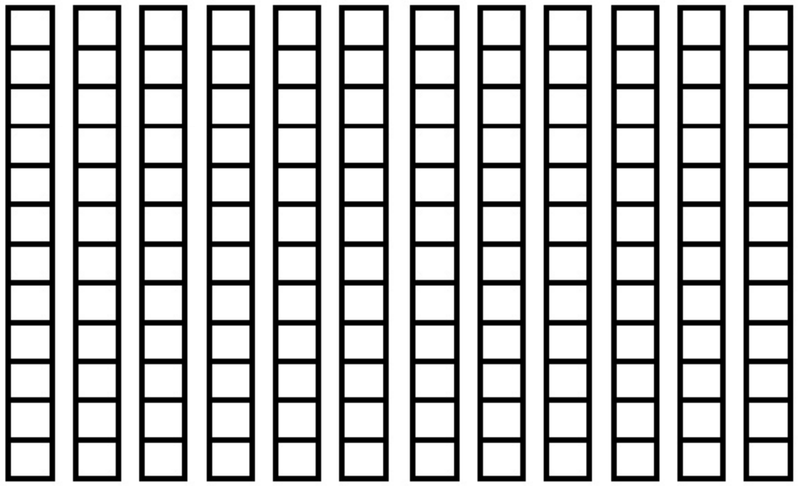
output feature vector

The **sequence-to-sequence** regression problem

output sequence



input sequence



The speech synthesis problem, as we currently understand it

- Input = linguistic features (phone identities, neighbours, + other context features)
  - can be represented as a (sparse) vector
  - this is our first encounter with a **distributed representation** of linguistic information

- Output = vocoder parameters

- Synthesis is then a **sequence to sequence regression problem** with two aspects:
  - regression from one feature set to the other
  - different "clock" rates of input to output features

HMM-based synthesis : training the models

- two *views*: regression *or* context-dependent modelling
- the regression view:
  - Sequencing (order of events, duration) = HMM topology + transition probabilities
  - Regression (input features to output features) = Regression tree
- the context-dependent modelling view:
  - construct a (v. large!) number of models, based on linguistic features
  - oops! most models have no training examples in the data!
  - solution: clusters of models for *similar* sounds, then have just one model for them all
  - this is exactly the same as the regression tree above (cluster = a leaf of the tree)

HMM-based synthesis : generating speech

- front end linguistic analysis
- flatten that to obtain a sequence of model names
- use the regression tree to obtain the models' parameters
- perform inference with the model = generate a sequence of frames
  - speech parameters (whatever the vocoder needs, such as MCCs, BAPs, F0)
  - use MLPG algorithm to ensure the parameters are smooth
- pass this to the vocoder which generates a speech waveform

What we will cover in this class

- Brief recap of video content and Q&A
- **Discussion points**

Comparison of some unit selection & SPSS synthesis samples

> Mini listening quiz: which is which? (and how to tell?!)

From text to speech with HMMs

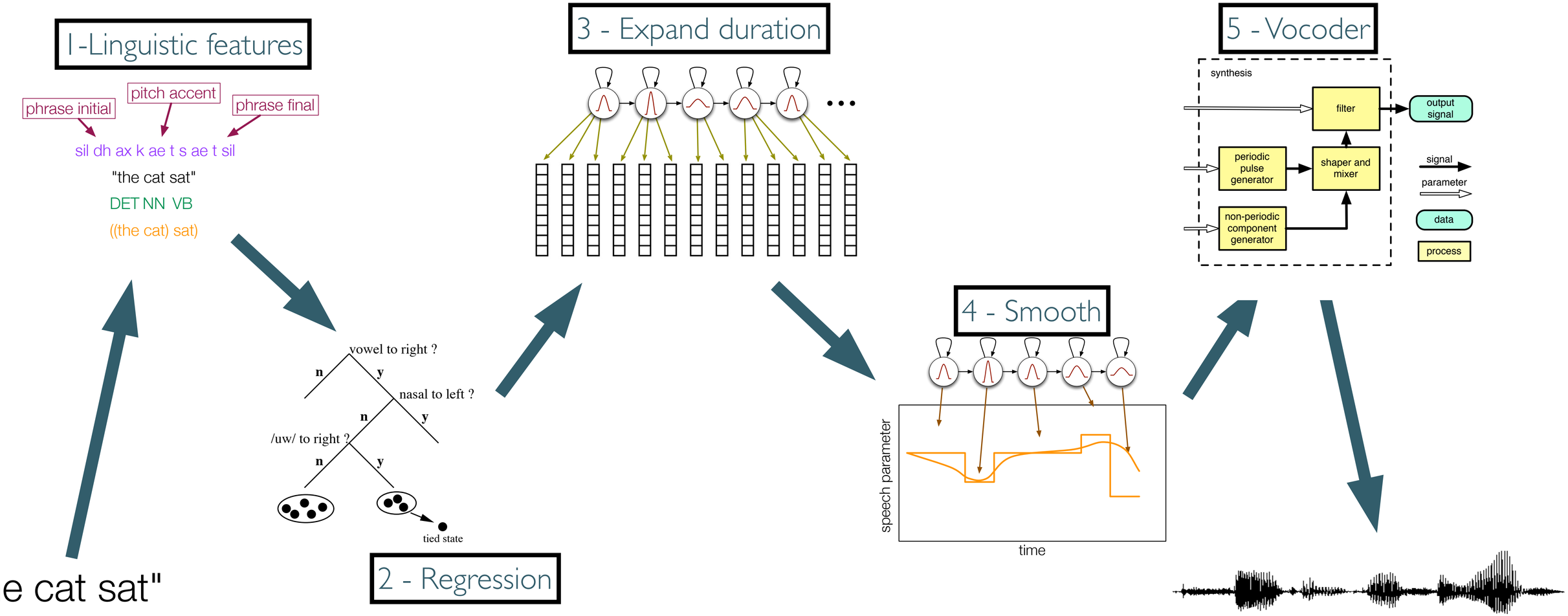
Q: What is the full sequence of steps from text to speech in HMM-based synthesis?

"the cat sat"

???

From text to speech with HMMs

Q: What is the full sequence of steps from text to speech in HMM-based synthesis?



The important role of context in TTS

We've talked a bit about context features, but let's think more about what their role is…

- Q: What would happen if we used no context features in unit selection? (i.e. only phone identity?)

- Q: And the same for HMM-based synthesis: what if we used few or no context features?

Controllability

Unit selection versus HMM-based synthesis

- Compare and contrast how the following could be realised in each method:

- Q: Make the voice speak faster or slower?
- Q: Make the voice speak in 5 different emotions?
- Q: Make the voice sound like a new person?

Unifying framework - sequence to sequence regression

- TTS is at heart a **sequence-to-sequence regression problem**
- So are **all** TTS methods, right up to the current State-of-the-Art

- Can you describe unit selection in terms of **sequence-to-sequence regression**?

Input **representation**

- representing features as **binary**
  - can this be done for **any** feature at all?
  - does this place any limitation on performance?

- **how and why** might you encode the following linguistic structures
  - place & manner of articulation
  - position of phone in syllable ; position of syllable in word ; position of word in phrase

- **upsample** all features to the acoustic **frame rate**
    - is this reasonable?

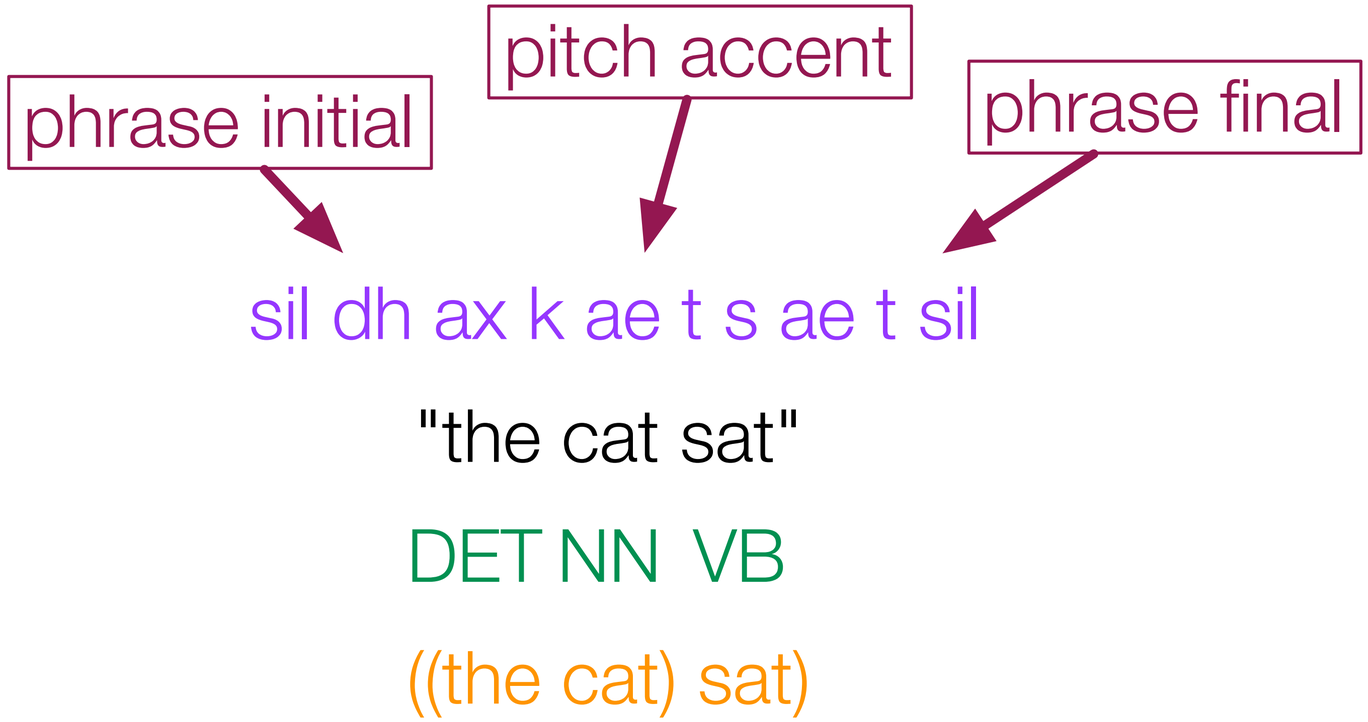
Exercise: a decision tree effectively treats the input features as "one hot"

- Draw a very simple decision tree that predicts the speech parameters for a phone
  - ignore duration for now - assume each phone has a duration of 1 frame
- Describe step-by-step how that can be used to predict a **sequence** of speech parameters
  - what are the **predictors** and what is the **predictee** ?

- List possible questions that could be asked in your decision tree
- Use your questions to rewrite the phone sequence as a sequence of one-hot vectors

- Draw a new decision tree that uses these vectors as the predictor

Exercise: a decision tree effectively treats the input features as "one hot"



What next?

- **Better regression model**
  - a Neural Network
  - input & output features essentially the same as regression tree + HMM

- Quality will still be limited by the **vocoder**

- Later, we will also address that problem
  - hybrid synthesis
  - direct waveform generation