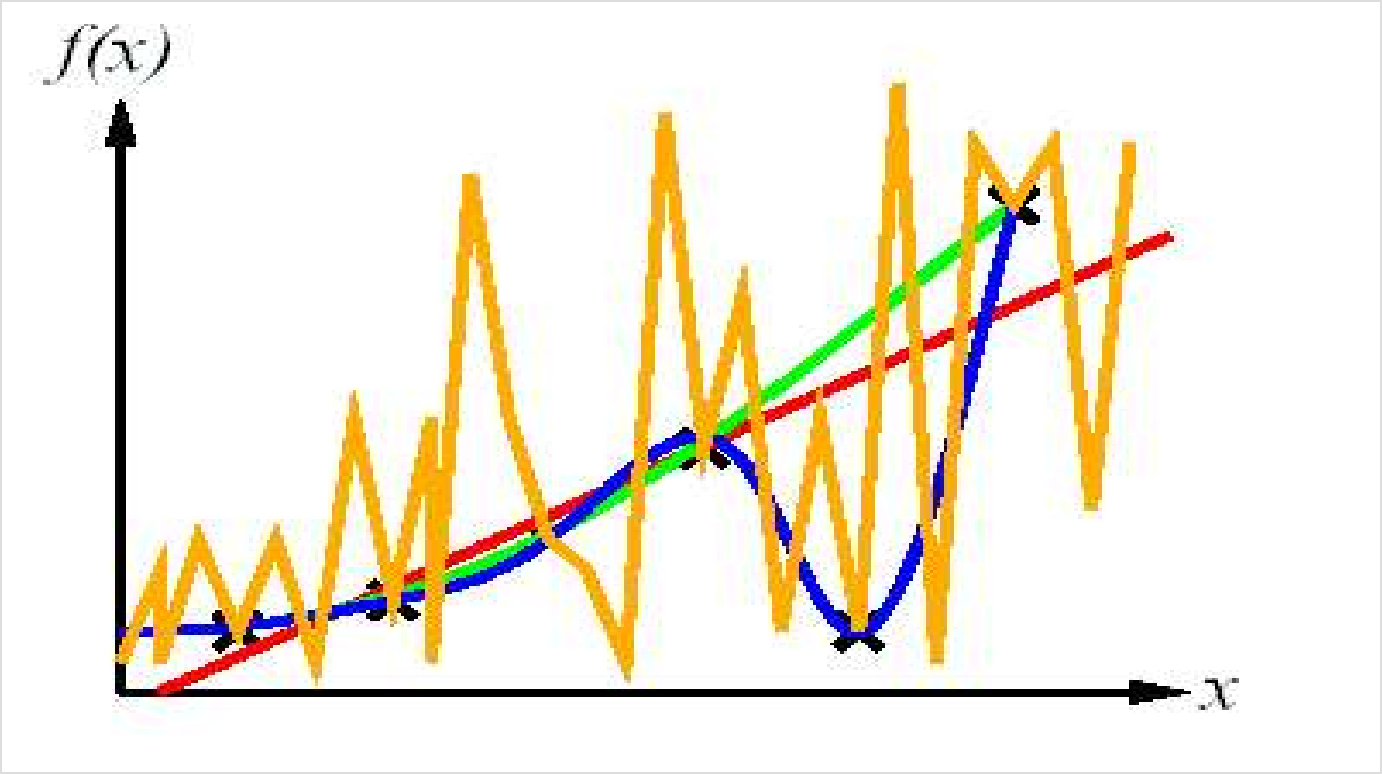
CSC421 Intro to Artificial Intelligence

UNIT 30:
Learning from observations

Inductive Learning Method

- Construct/adjust h to agree with f on training set e.g curve fitting



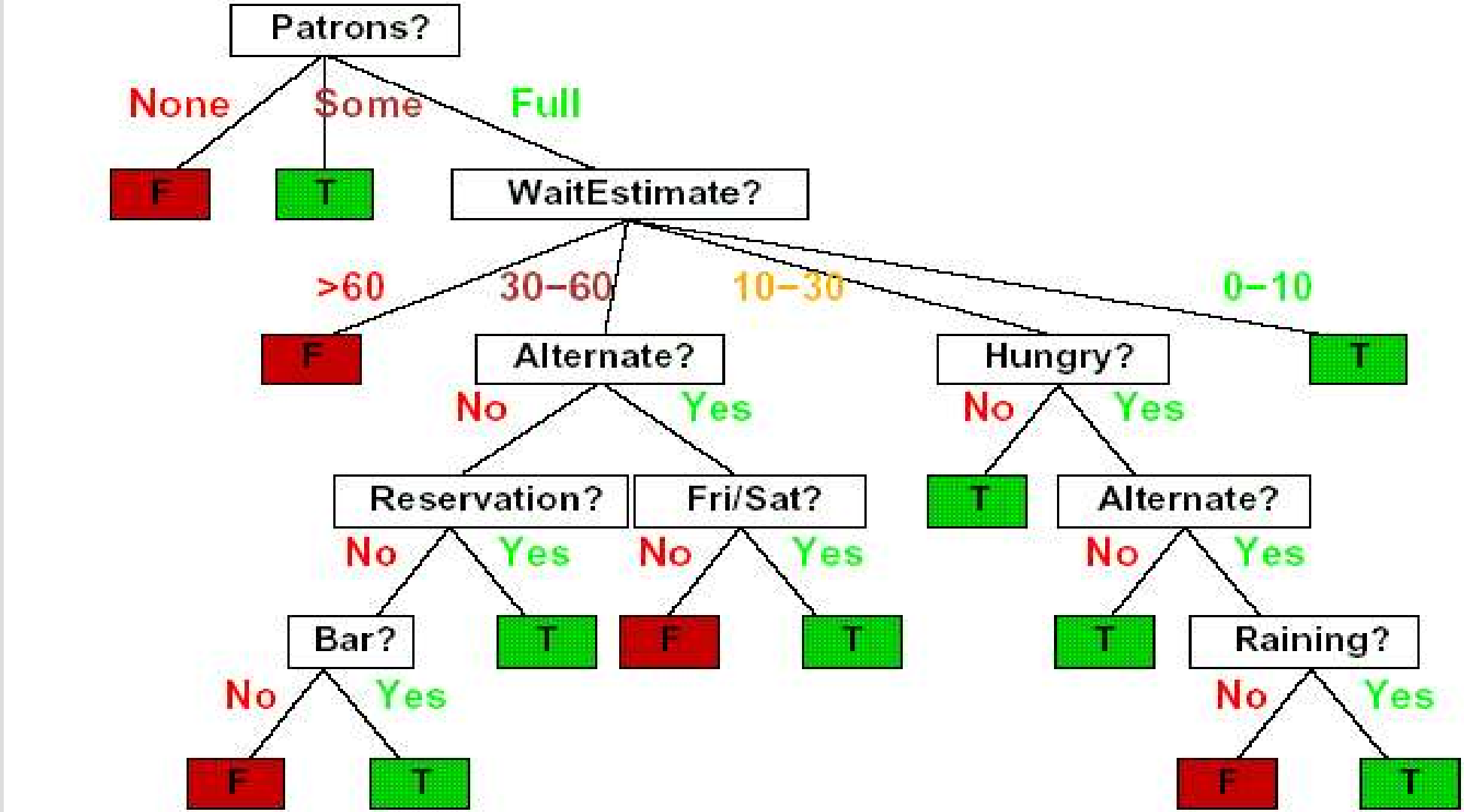
CONSISTENT HYPOTHESIS

OCCAM'S RAZOR
Maximize a combination of consistency and simplicity

Attribute-value representations

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

Decision Tree



Expressiveness

- Decision trees can express any function of the input attributes
  - Truth table row -> path from to leaf
- Trivially there is consistent decision tree for any training set w/ one path to leaf for each example but it probably won't generalize to new examples
- Prefer to find more compact decision trees

Hypothesis spaces

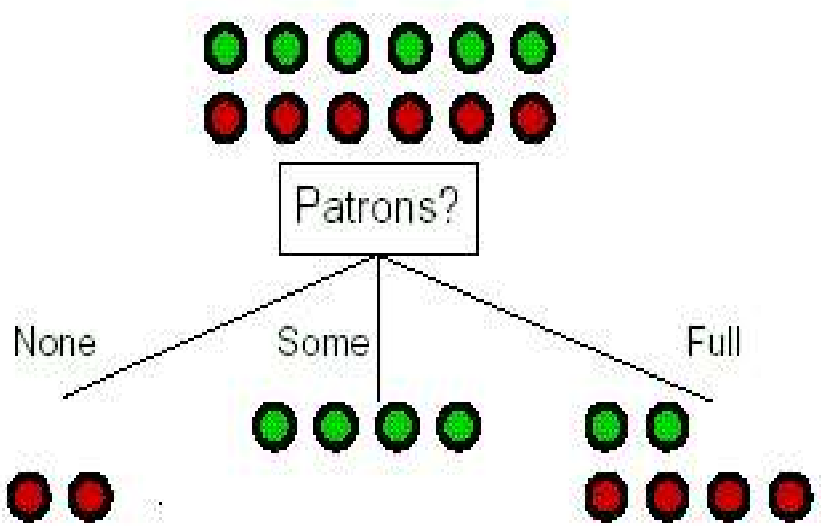
- How many distinct decision trees with n boolean attributes ?
  - = number of boolean functions
  - = number of distinct truth tables with 2^n rows = ?
- For 6 boolean attributes there are only 18,446,744, 073, 709,551,616 trees

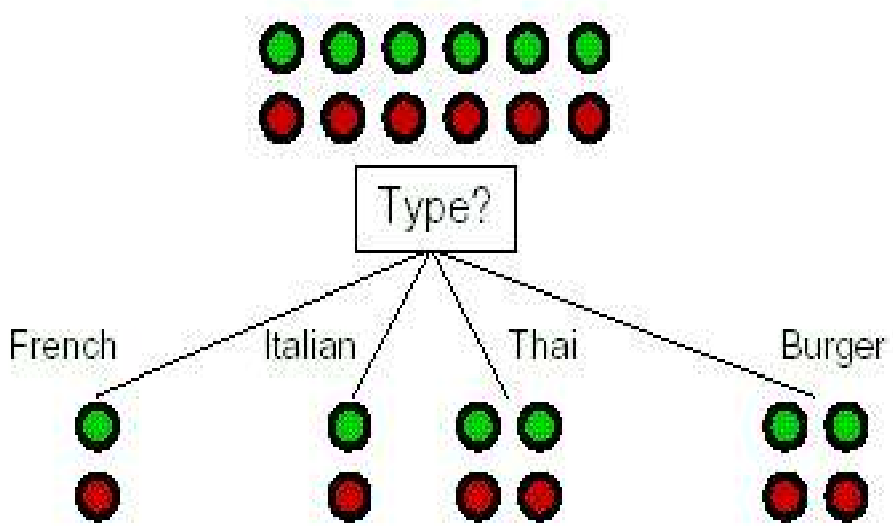
Decision Tree Learning

- Aim: Find a small tree consistent with training examples
- Idea: (recursively) choose the “most significant” attribute as root of the subtree

Choosing an attribute

- Idea: a good attribute splits examples into subsets that are "ideally" all + or -
- Patrons? is better choice – gives more "information" about classification





Information

- Information answers questions
- Scale = 1 bit = answer to Boolean question with prior <0.5, 0.5>
- Information in an answer where prior is $\langle P_1,\ldots,P_n\rangle$ is
- $H(\langle P_1,\ldots,P_n\rangle) = \Sigma_{i=1}^{n} -P_i \log_2 P_i$
- Also called entropy of the prior

Historical Sidenote

- Claude E. Shannon – Information theory
- 1948 – Most efficient way to encode a message
- Other: juggling, unicycling
- Inventions:
  - Rocket powered pogo stick
  - Theseus mouse

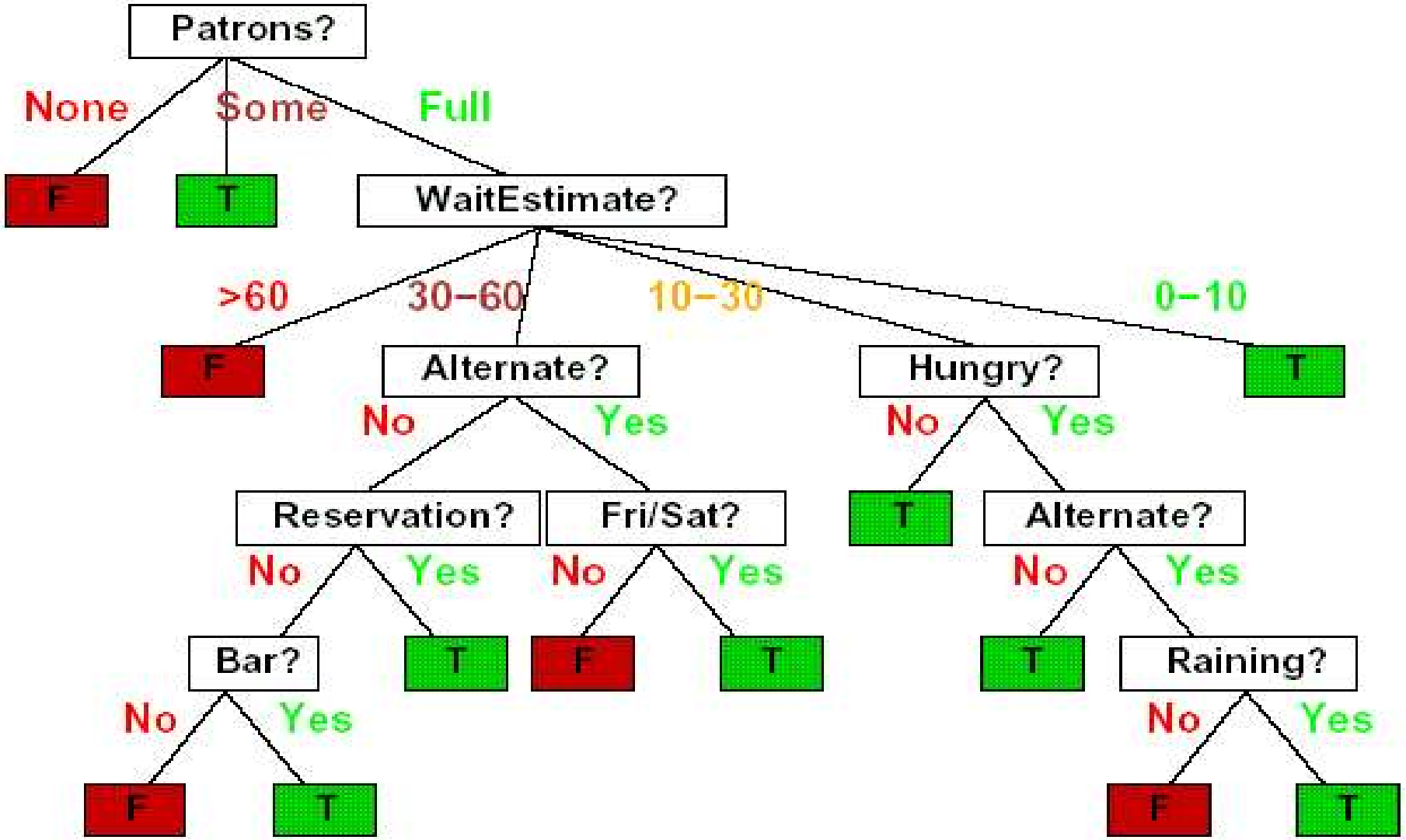
Information cont'd

- Suppose we have p positive and n negative examples at the root then:
  - H(< p / p+n, n / p+n>) bits to classify new
- An attribute splits the examples E into subsets E_i, each of which (we hope) needs less information to complete the classification

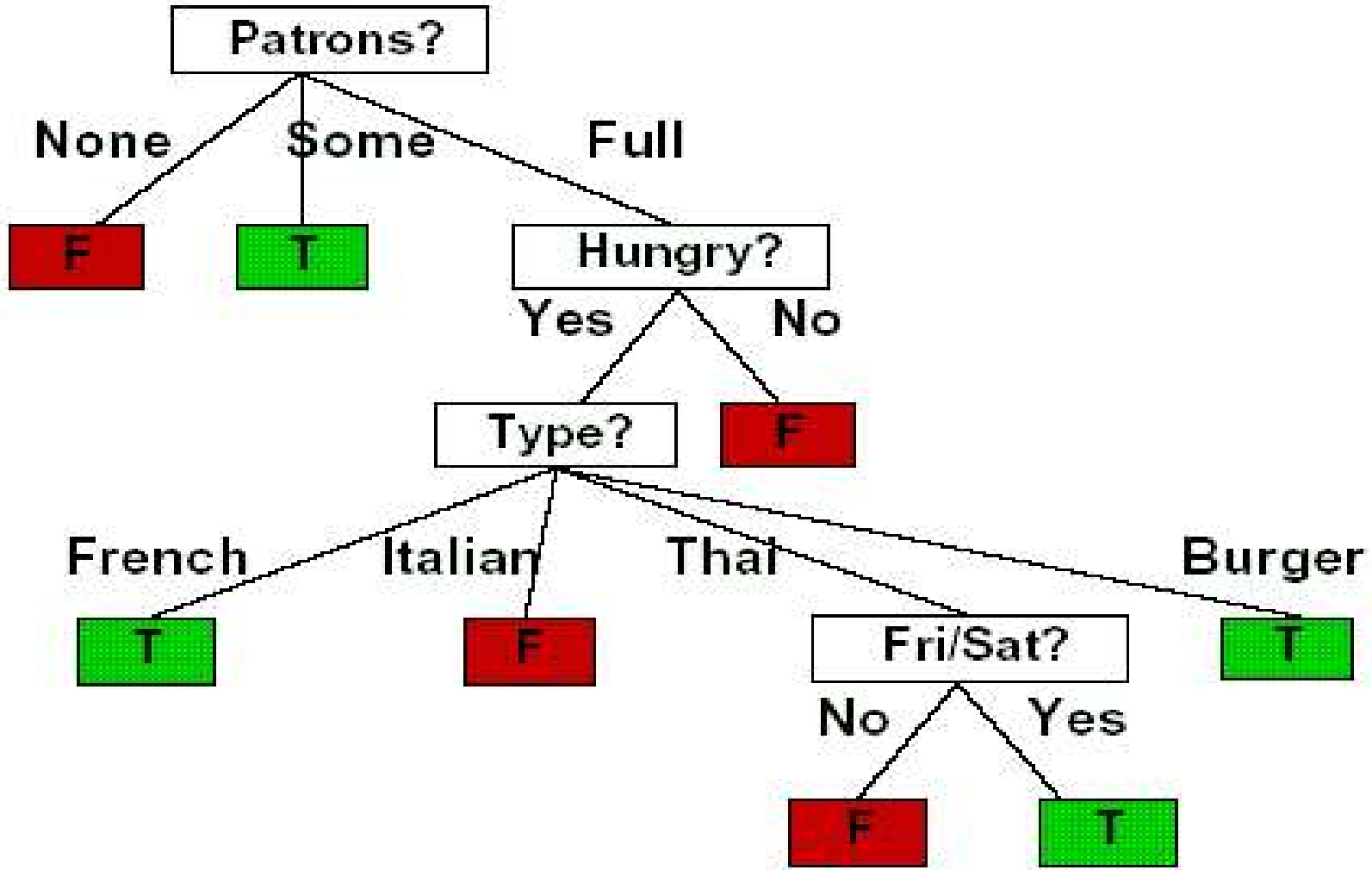
Information Gain

- Let E_i have p_i positive and n_i negative
  - $H(<p_i / (p_i + n_i), n_i/(p_i + n_i)>)$ bits needed to classify a new example
  - Expected # of bits per example for all branches:
  - $\Sigma_i \; p_i + n_i / (p_i + n_i) \;\; H(<p_i/(p_i + n_i), \; n_i/(p_i + n_i)>)$
- For Patrons this is 0.459, for Type still 1 bit
  - Choose the attribute that minimizes the remaining information needed

“Default” decision tree



Learned decision tree



Performance measurement

- How do we know $h \approx f$?
  - Theorems of learning theory
  - Try h on "new" set of training samples
- Learning curve = % correct as f(training size)