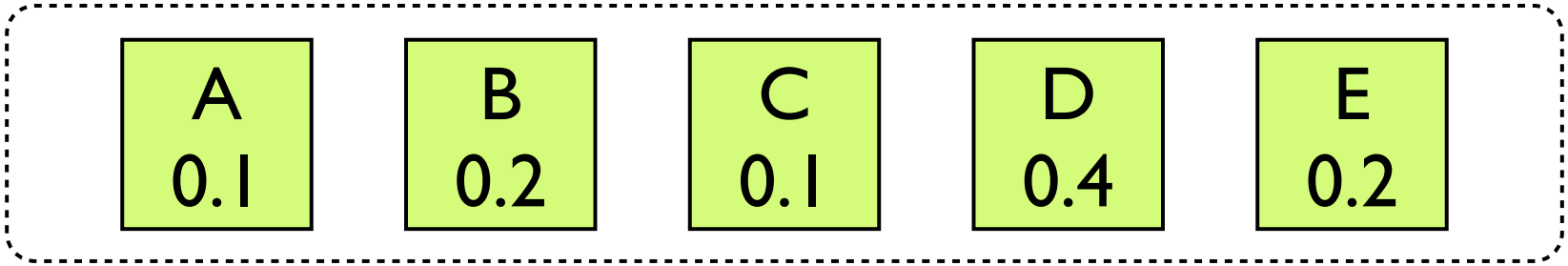
TPPmark2012

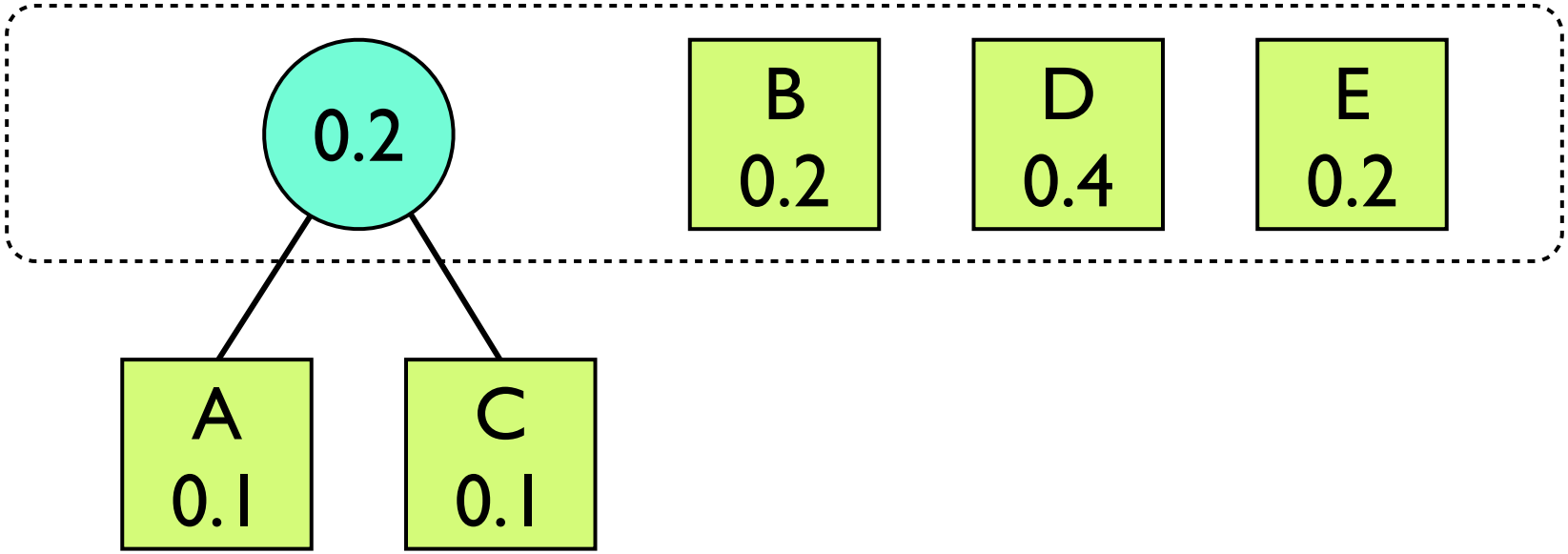
問題説明

ハフマン木の構成

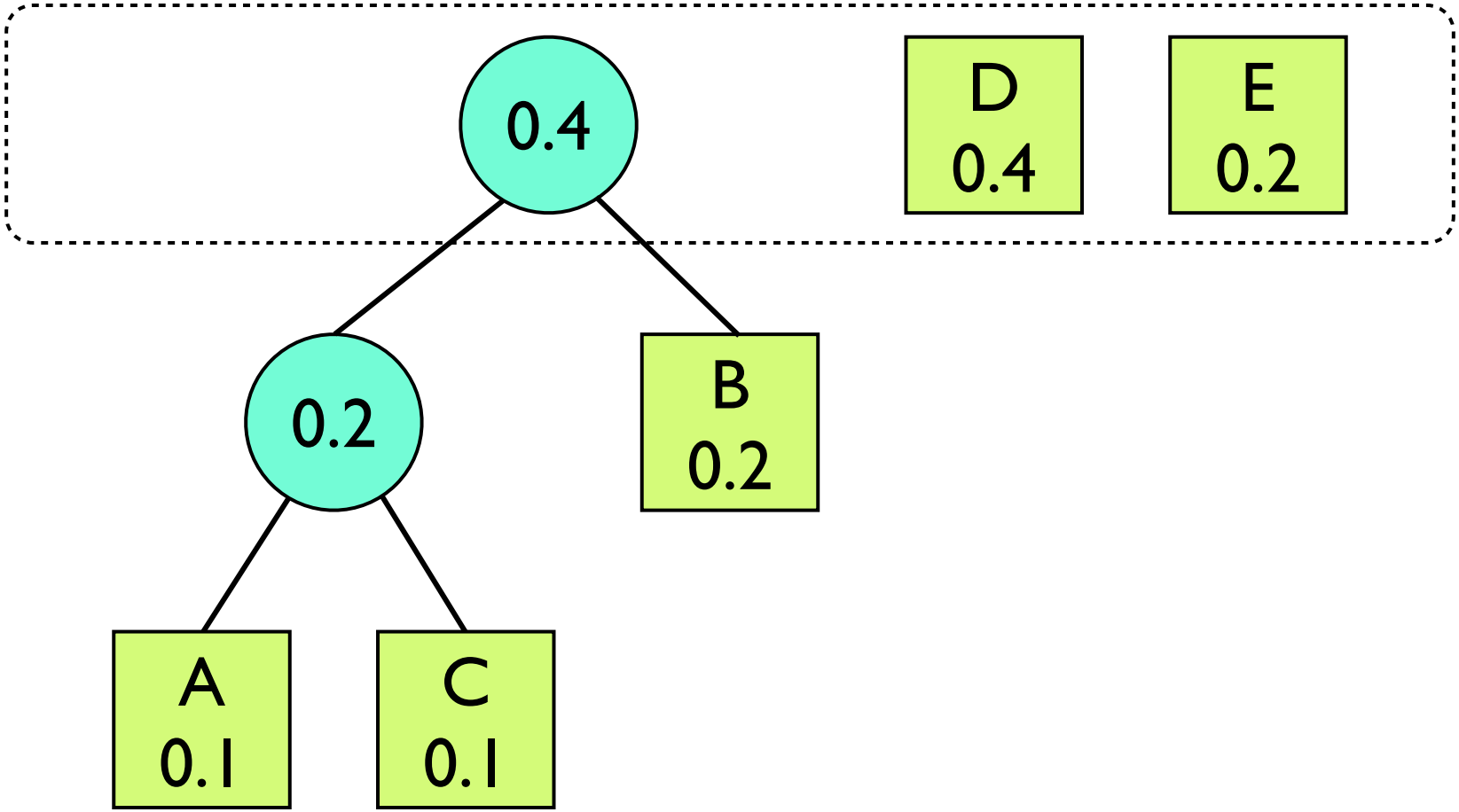
ハフマン木の構成



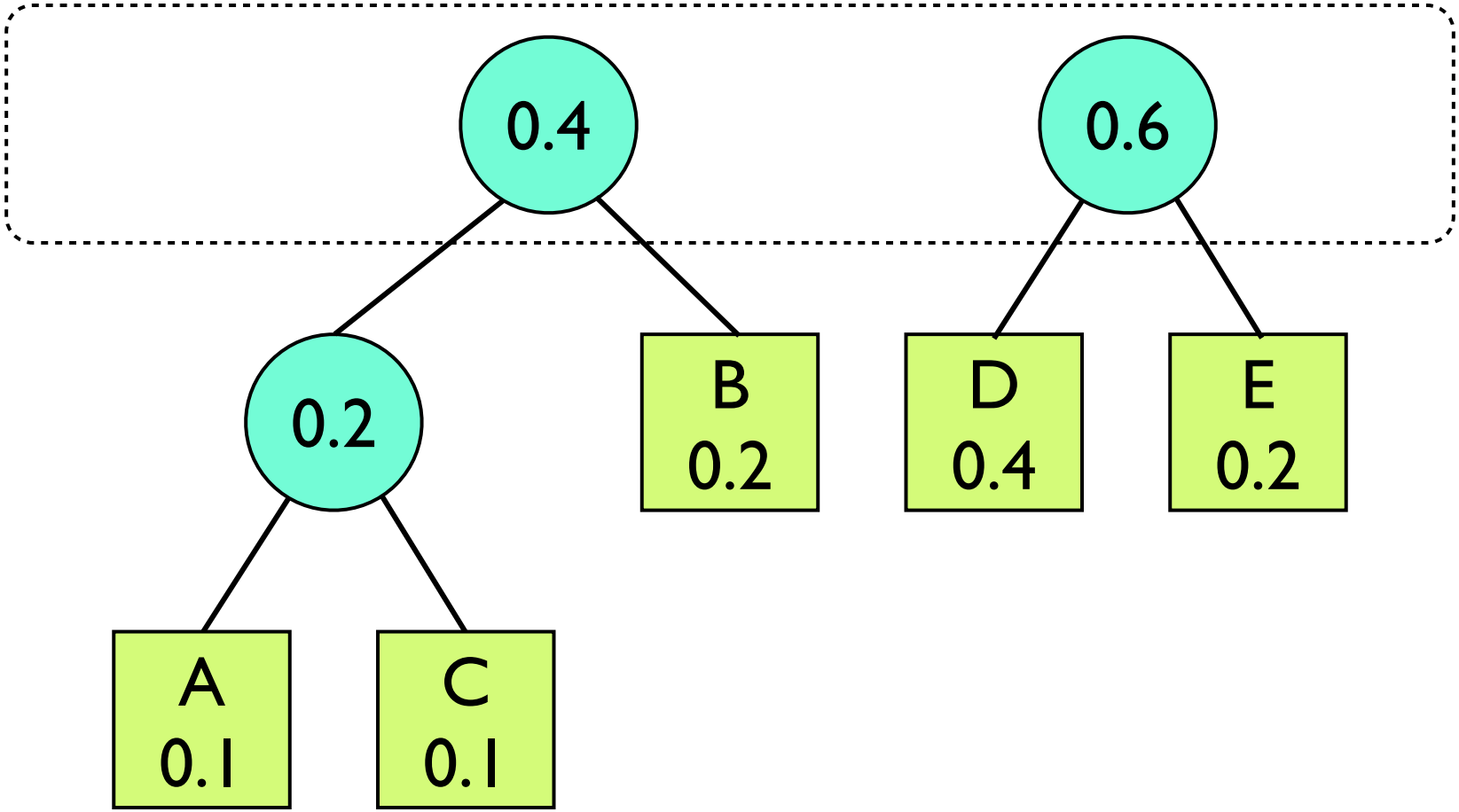
ハフマン木の構成



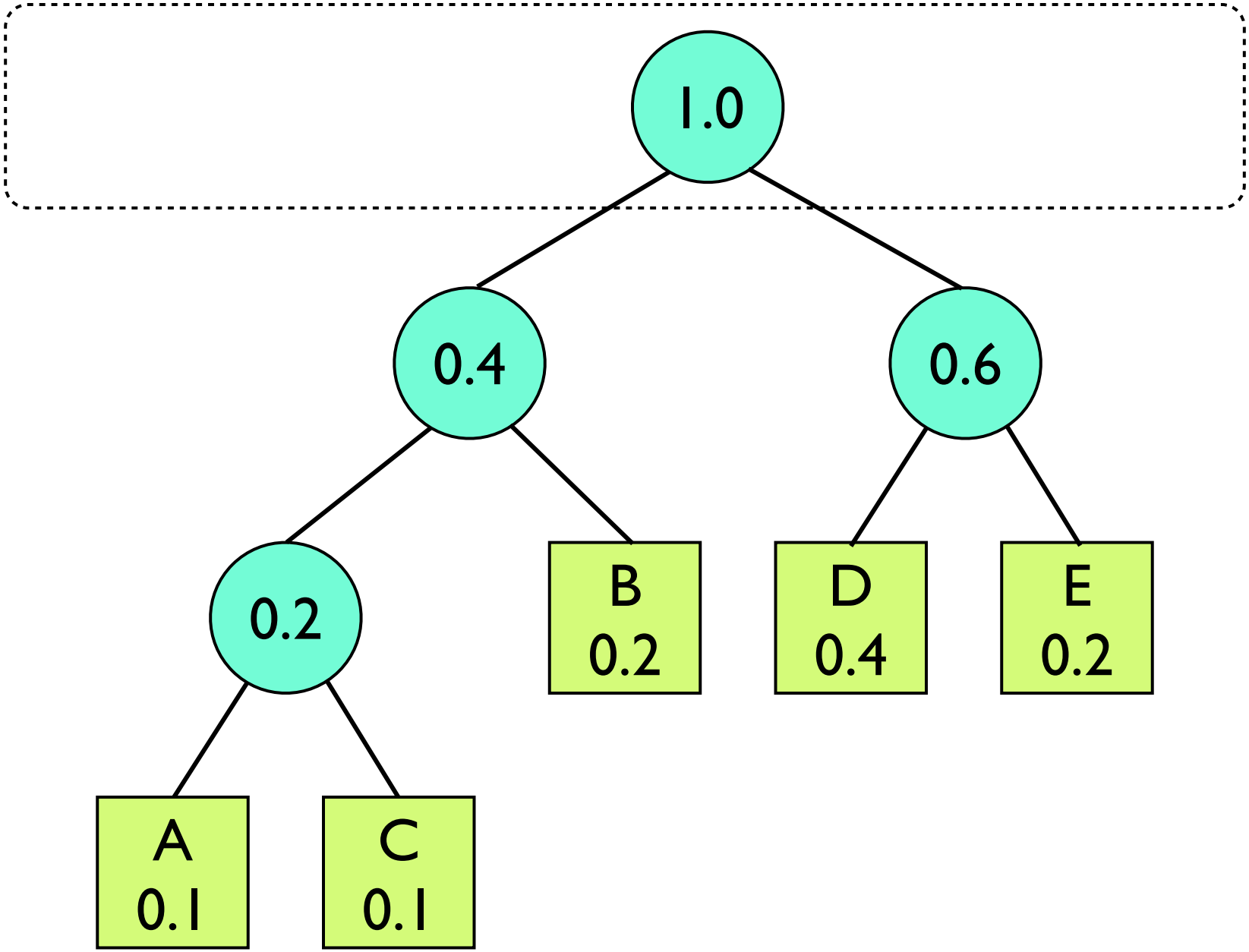
ハフマン木の構成



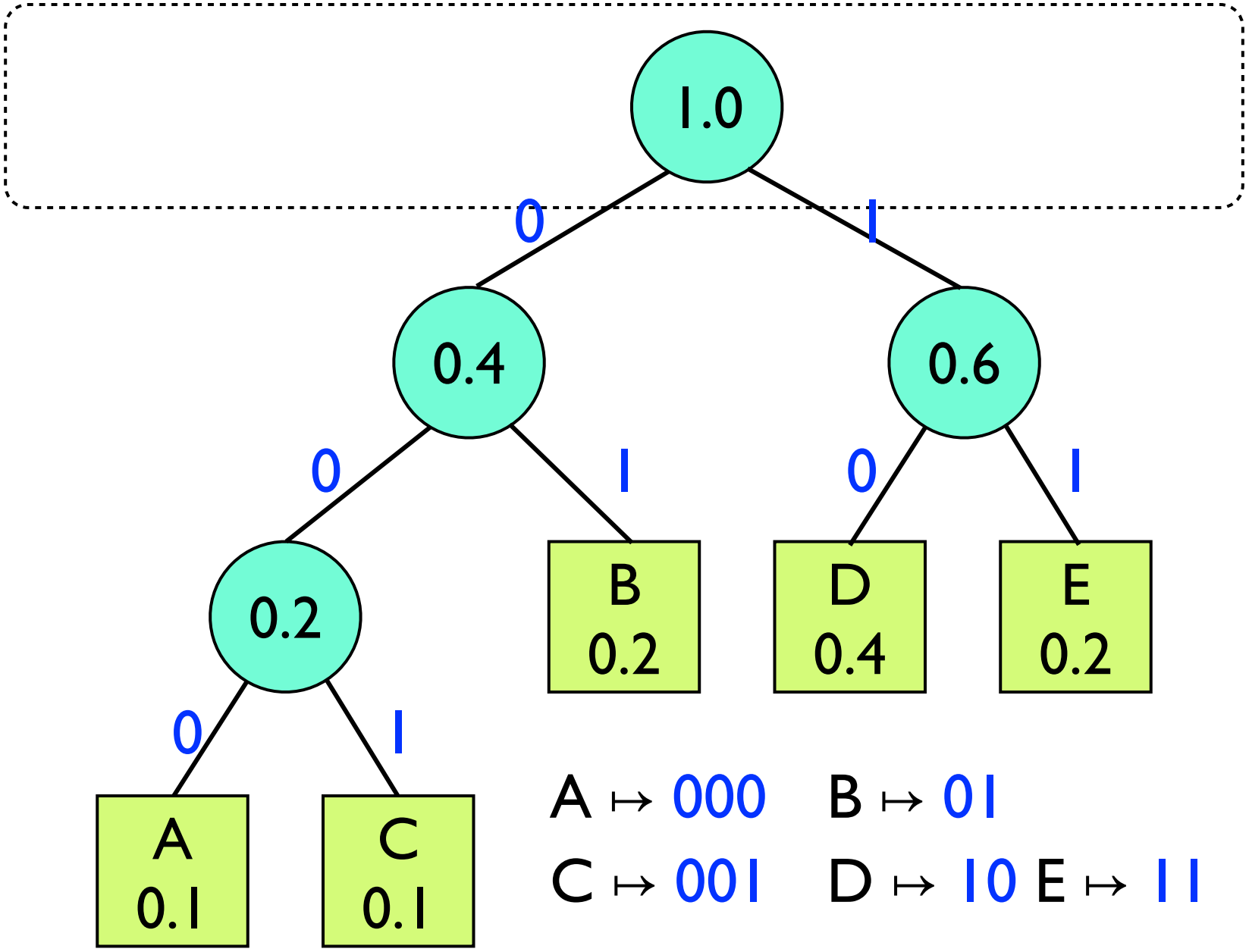
ハフマン木の構成



ハフマン木の構成



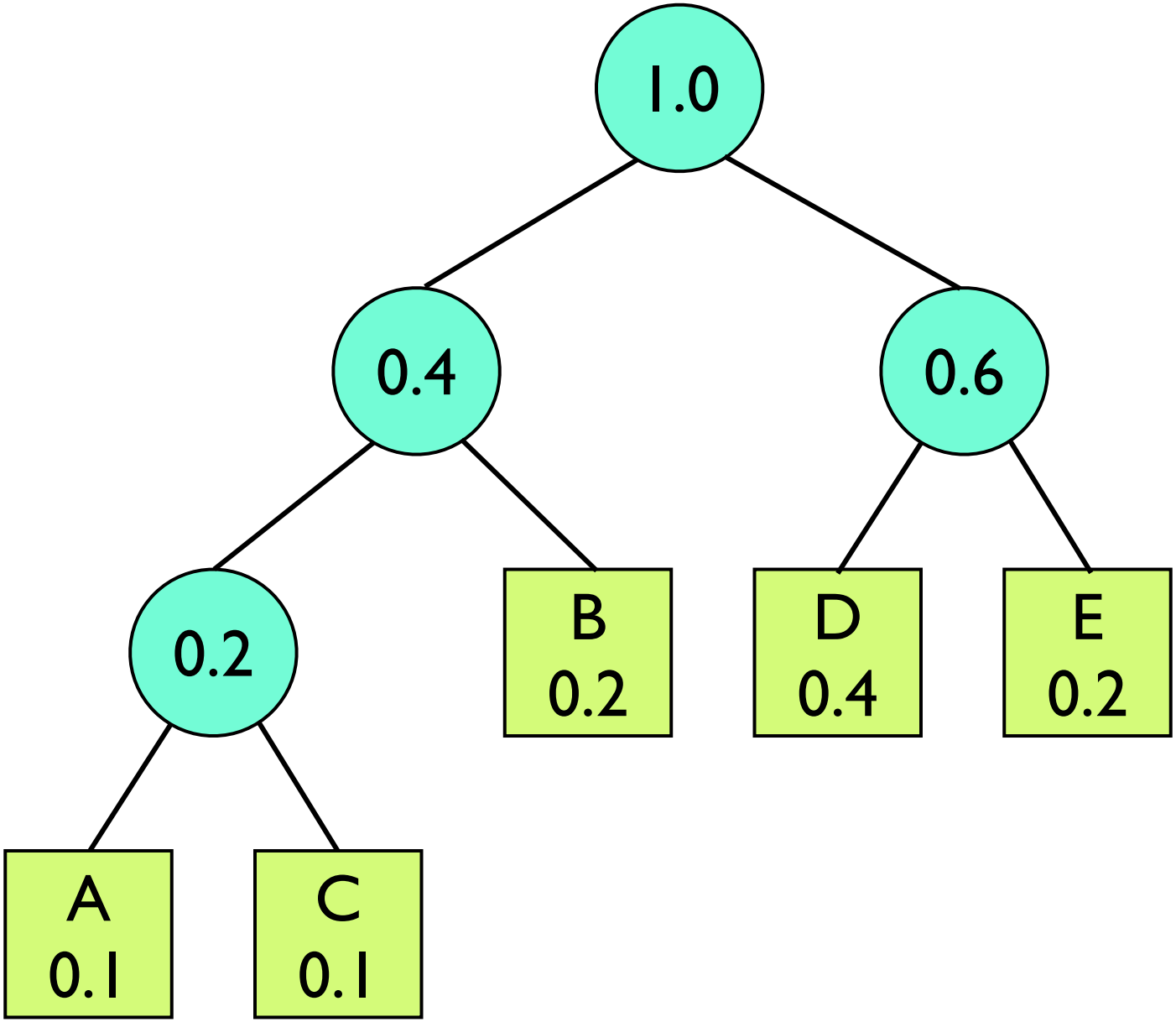
ハフマン木の構成



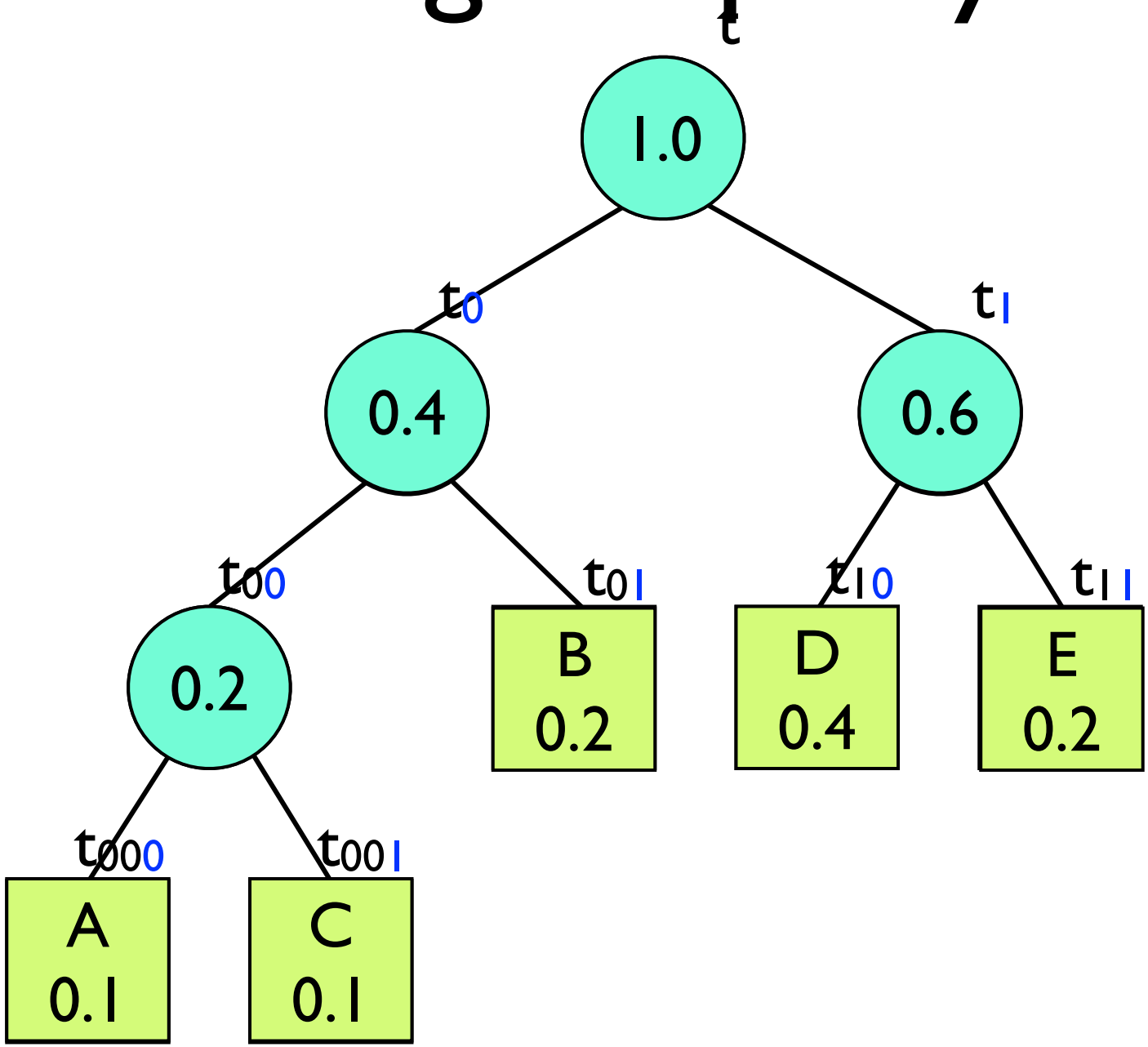
A ↦ 000　B ↦ 01

C ↦ 001　D ↦ 10 E ↦ 11

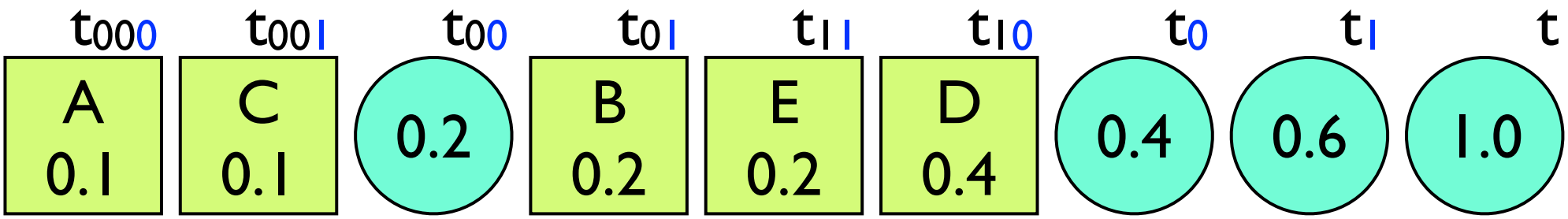
Sibling Property



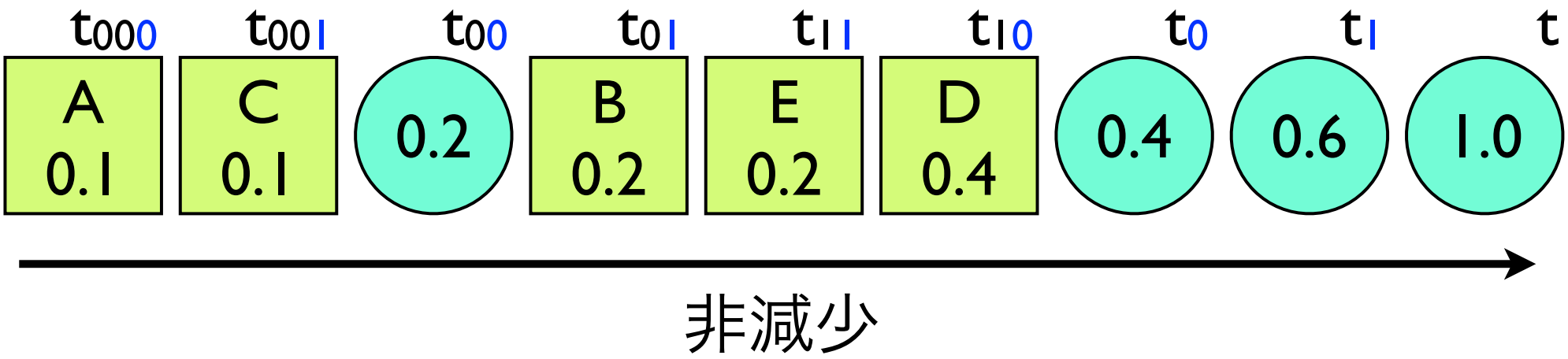
Sibling Property



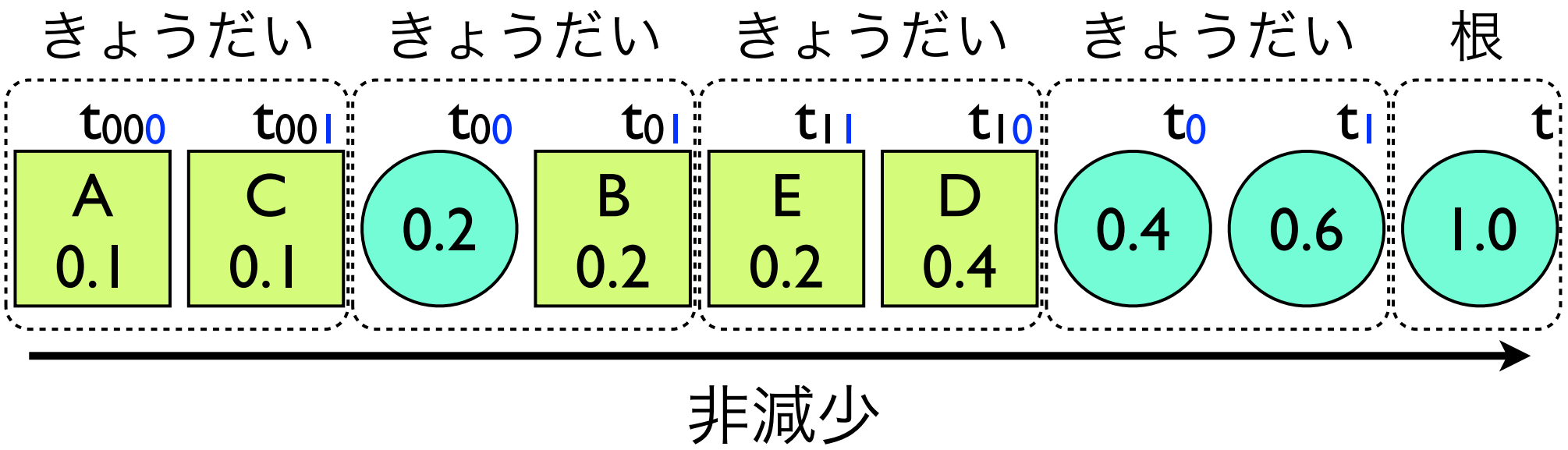
Sibling Property



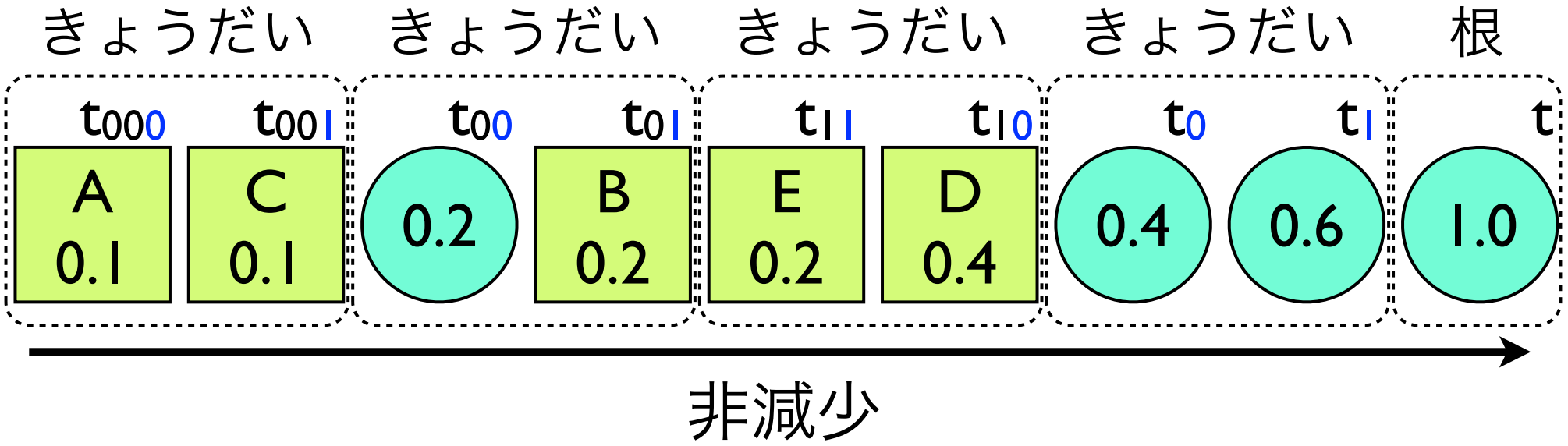
Sibling Property



Sibling Property



Sibling Property



- Sibling Property ... 符号の木のノードを、このように一列に並べることができるという性質

ハフマン木とSibling Property

[Gallager 78]

- 符号の木について、以下の２つは同値
  1. ハフマン木である
     - アルゴリズムによる特徴付け
     - 静的ハフマン符号向け
  2. Sibling Propertyを持つ
     - 構造的特徴付け
     - 動的ハフマン符号向け
- TPPmark2012
  - 1⇒2 と 2⇒1 をそれぞれ示せ

Informal Proof [Gallager 78]

Proof: First assume that a code tree has the sibling property. Then the last two elements on the ordered list are siblings and, in addition, must be leaves, for if one were an intermediate node, at least one of the children of that intermediate node would have a smaller probability than the intermediate node,[1] which is impossible because of the ordering property. Thus these nodes correspond to two smallest probability source letters and can be made siblings in the first execution of step 2) in the Huffman algorithm. Now remove these siblings from the code tree, removing also the last two elements from the ordered list. The resulting reduced code tree still has the sibling property, and the leaves of the reduced code tree correspond to the list L in the Huffman algorithm after the first execution of step 3). Thus the above argument can be repeated: at each step the Huffman algorithm chooses, as siblings, elements which are siblings in the original code tree. By matching the link labels in the Huffman code to those in the original code tree, the two codes are seen to be identical.

[1]This is where we use the assumption that at most one source letter have zero probability. The theorem is true without this restriction, but the proof is harder and the restriction is of no importance.

← 2⇒1 (19行)

1⇒2 (13行)

↓

Next assume that a binary code tree is generated by the Huffman algorithm, and assume that each time the algorithm executes step 2), we add the two nodes defined as siblings to the top of an initially empty list, putting the less probable below the more probable. The list so generated clearly has each node adjacent to its sibling, so to establish the sibling property, we simply have to show that the list is nonincreasing in order of probability. This is trivial, however, since at each iteration the two elements added to the list have probabilities less than or equal to that of each element in the new L of the Huffman algorithm, and the next two elements added to the list are chosen from this new L. Q.E.D.

容易に見える点

- informal proofが短い
- 使っているデータ構造は標準的
  - 二分木や(ソートされた)リスト
  - 出現確率は頻度にすれば正整数
- 「符号の木である」の記述は簡単

困難が予想される点

- 符号の木が「ハフマン木である」ことをどう記述するか
  - 構成アルゴリズム: 葉の集合が与えられたとき、ボトムアップにハフマン木(のひとつ)を作る方法
  - 欲しいもの: 符号の木が与えられたとき、それがハフマン木であることを表す述語

先行研究: Formalising Huffman's algorithm [Théry 2004] (on Coq)

- ハフマン符号が optimal であることの形式的証明
- Sibling Propertyとの関連は述べていない
- パスを陽に出さない形式化
  - Coverという概念

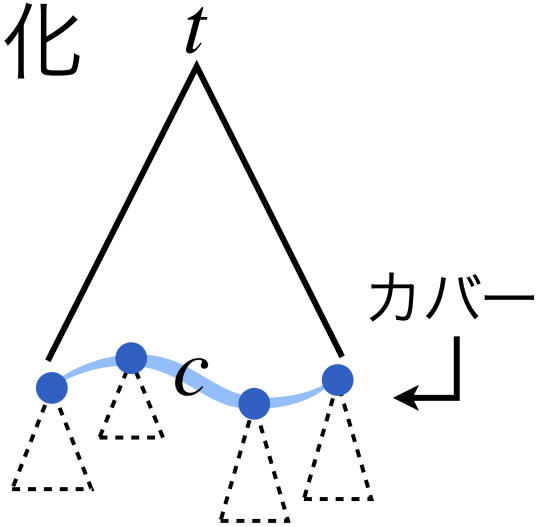
主要な概念: Cover

カバー：葉全体の一般化

cover c t：

c は二分木 t のカバー

c の要素は t の“葉”



葉全体はカバー:

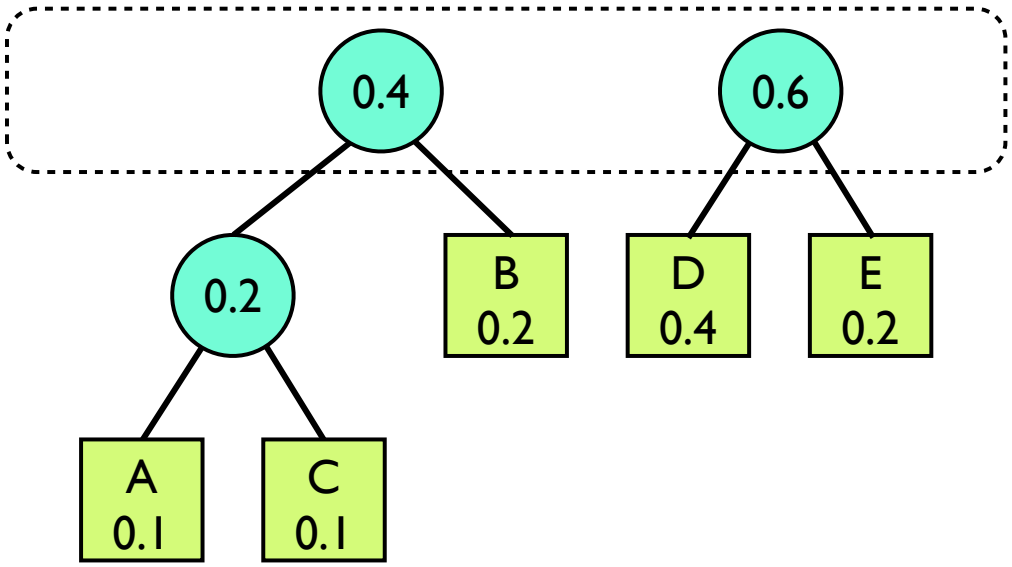
$$\text{cover}\ (\mathit{leaves}\ t)\ t$$

カバーは並べ替えを許す:

$$c \approx c' \wedge \text{cover}\ c\ t \Rightarrow \text{cover}\ c'\ t$$

Cover

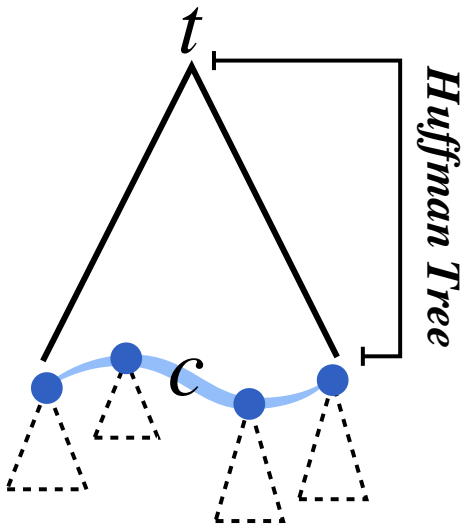
- 重なりのない部分木からなるリストで、本来の葉が部分木のどれかに含まれるもの
- ハフマン木構成の各段階で現れるものは、最終的なハフマン木のカバーとして表現できる



ハフマン木であること

build c t :

t は c を“葉全体”とするハフマン木

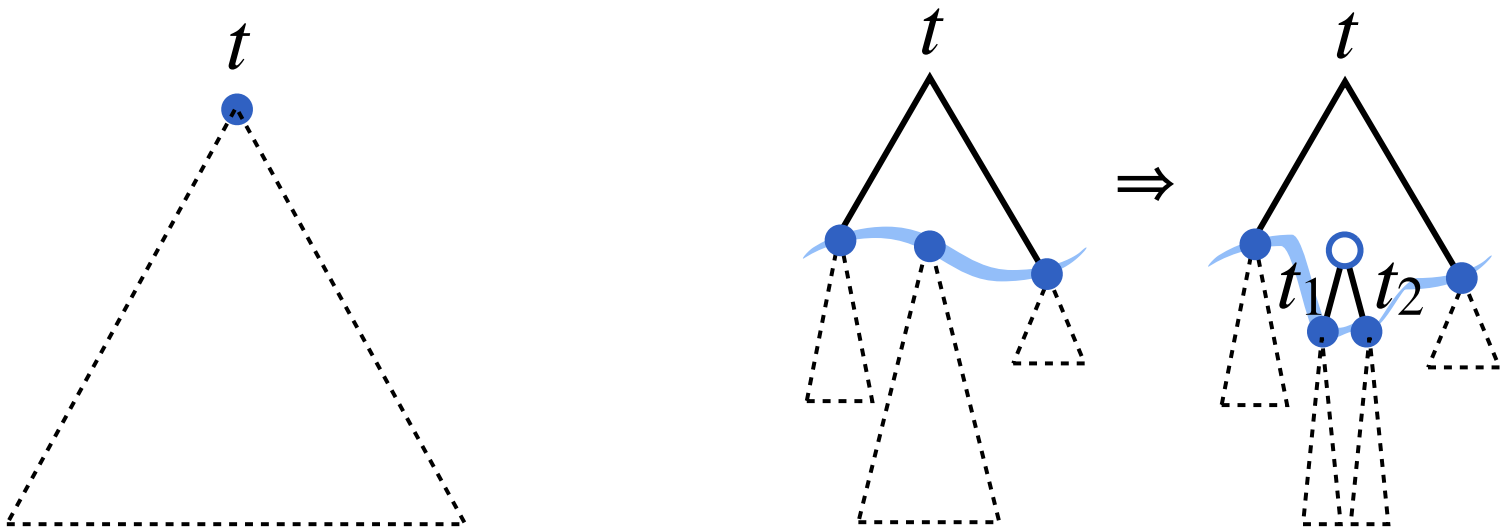


$c = leaves\ t$ のとき、通常のハフマン木

build c t ⇒ cover c t

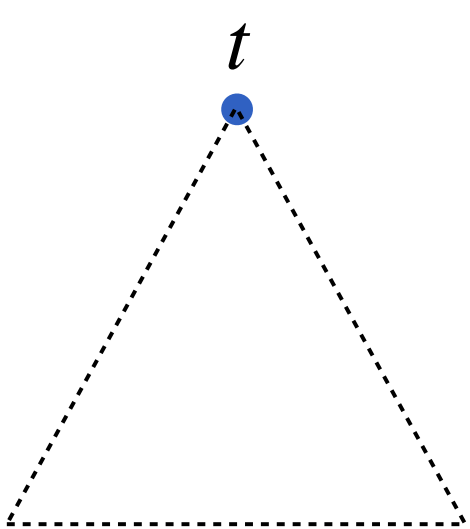
Coverの帰納的定義

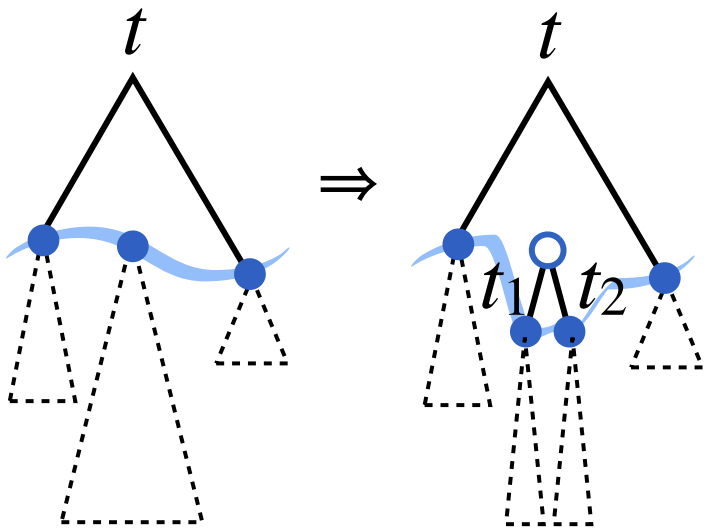
$$\frac{}{\text{cover } [t]\ t} \qquad \frac{c \approx [t_1; t_2] \# c_1 \quad \text{cover } ([t_1 \wedge t_2] \# c_1)\ t}{\text{cover } c\ t}$$



Buildの帰納的定義

$$\frac{}{\mathsf{build}\ [t]\ t} \qquad \frac{\mathsf{occ}\ t_1 \leq \mathsf{occ}\ t_2 \quad \forall t' \in c_1.\ \mathsf{occ}\ t_2 \leq \mathsf{occ}\ t' \qquad c \approx [t_1; t_2] \# c_1 \quad \mathsf{build}\ ([t_1 \wedge t_2] \# c_1)\ t}{\mathsf{build}\ c\ t}$$





定理証明系Coqを用いたSibling Propertyとハフマン木との同値性の形式的証明[須田, 山本]

- PPL 2012 ポスターセッション
- Théryの形式化に続く形
  - パスを陽に扱わない
- Cover, Buildをそれぞれ拡張した Wrapper, Wbuildを定義
- 詳しくは須田氏から...