Prüfer codes and Cayley's formula

Math 40210, Fall 2015

November 3, 2015

The Prüfer code of a tree

The Construction

- Given: a tree T on vertex set $\{v_1, \ldots, v_n\}$, with $v_1 < v_2 \ldots < v_n$
- Repeat until only one edge left:
  - Delete leaf with lowest label (result is smaller tree)
  - Record the label of the deleted leaf's unique neighbor
- Result is a string of length $n - 2$ on alphabet $\{v_1, \ldots, v_n\}$, the *Prüfer code* of T

The Prüfer code of a tree

The Construction

- Given: a tree T on vertex set $\{v_1, \ldots, v_n\}$, with $v_1 < v_2 \ldots < v_n$
- Repeat until only one edge left:
  - Delete leaf with lowest label (result is smaller tree)
  - Record the label of the deleted leaf's unique neighbor
- Result is a string of length $n-2$ on alphabet $\{v_1, \ldots, v_n\}$, the *Prüfer code* of T

Some Facts

- FACT 1 (fairly easy): Once first leaf (say v_i) is deleted, and first label (say v_j) recorded, rest of Prüfer code is exactly the Prüfer code of $T - v_i$ on vertex set $\{v_1, \ldots, v_n\} \setminus \{v_i\}$

The Prüfer code of a tree

The Construction

- Given: a tree T on vertex set $\{v_1, \ldots, v_n\}$, with $v_1 < v_2 \ldots < v_n$
- Repeat until only one edge left:
  - Delete leaf with lowest label (result is smaller tree)
  - Record the label of the deleted leaf's unique neighbor
- Result is a string of length $n - 2$ on alphabet $\{v_1, \ldots, v_n\}$, the *Prüfer code* of T

Some Facts

- FACT 1 (fairly easy): Once first leaf (say v_i) is deleted, and first label (say v_j) recorded, rest of Prüfer code is exactly the Prüfer code of $T - v_i$ on vertex set $\{v_1, \ldots, v_n\} \setminus \{v_i\}$
- FACT 2 (follows by induction from FACT 1): Each vertex v_i appears $d(v_i) - 1$ times in the Prüfer code

Different trees have different Prüfer codes

Proof by Induction

- Case $n = 3$ easily verified

Different trees have different Prüfer codes

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given trees $T_1,\ T_2$ on $\{v_1, \ldots, v_n\}$
  - IF lowest-labeled leaves are different, then the Prüfer codes are different (by FACT 2)

Different trees have different Prüfer codes

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given trees T_1, T_2 on $\{v_1, \ldots, v_n\}$
  - IF lowest-labeled leaves are different, then the Prüfer codes are different (by FACT 2)
  - IF lowest-labeled leaves the same, but labels of unique neighbours different, THEN the Prüfer codes are different (by construction)

Different trees have different Prüfer codes

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given trees T_1, T_2 on $\{v_1, \ldots, v_n\}$
  - IF lowest-labeled leaves are different, then the Prüfer codes are different (by FACT 2)
  - IF lowest-labeled leaves the same, but labels of unique neighbours different, THEN the Prüfer codes are different (by construction)
  - IF lowest-labeled leaves the same, labels of unique neighbours the same, THEN step one of Prüfer code constructions agree; but resulting smaller trees are different, so have different Prüfer codes (by induction)

Different trees have different Prüfer codes

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given trees T_1, T_2 on $\{v_1, \ldots, v_n\}$
    - IF lowest-labeled leaves are different, then the Prüfer codes are different (by FACT 2)
    - IF lowest-labeled leaves the same, but labels of unique neighbours different, THEN the Prüfer codes are different (by construction)
    - IF lowest-labeled leaves the same, labels of unique neighbours the same, THEN step one of Prüfer code constructions agree; but resulting smaller trees are different, so have different Prüfer codes (by induction)

Map from Trees to Prüfer Codes is *Injective*

Every string is the Prüfer code of some tree

Proof by Induction

- Case $n = 3$ easily verified

Every string is the Prüfer code of some tree

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given string $S = (\sigma_1, \ldots, \sigma_{n-2})$ on alphabet $\{v_1, \ldots, v_n\}$
  - Find the lowest labeled vertex that does not appear in the string, v_i say
  - Form $S' = (\sigma_2, \ldots, \sigma_{n-2})$ by deleting first entry from S

Every string is the Prüfer code of some tree

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given string $S = (\sigma_1, \ldots, \sigma_{n-2})$ on alphabet $\{v_1, \ldots, v_n\}$
  - Find the lowest labeled vertex that does not appear in the string, v_i say
  - Form $S' = (\sigma_2, \ldots, \sigma_{n-2})$ by deleting first entry from S
  - S' is a string of length $n - 3$ on alphabet $\{v_1, \ldots, v_n\} \setminus v_i$, so (by induction) there is a tree T' on $\{v_1, \ldots, v_n\} \setminus v_i$ with S' as its Prüfer code

Every string is the Prüfer code of some tree

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given string $S = (\sigma_1, \ldots, \sigma_{n-2})$ on alphabet $\{v_1, \ldots, v_n\}$
  - Find the lowest labeled vertex that does not appear in the string, v_i say
  - Form $S' = (\sigma_2, \ldots, \sigma_{n-2})$ by deleting first entry from S
  - S' is a string of length $n - 3$ on alphabet $\{v_1, \ldots, v_n\} \setminus v_i$, so (by induction) there is a tree T' on $\{v_1, \ldots, v_n\} \setminus v_i$ with S' as its Prüfer code
  - Form T from T' by adding vertex v_i, joined only to σ_1

Every string is the Prüfer code of some tree

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given string $S = (\sigma_1, \ldots, \sigma_{n-2})$ on alphabet $\{v_1, \ldots, v_n\}$
  - Find the lowest labeled vertex that does not appear in the string, v_i say
  - Form $S' = (\sigma_2, \ldots, \sigma_{n-2})$ by deleting first entry from S
  - S' is a string of length $n - 3$ on alphabet $\{v_1, \ldots, v_n\} \setminus v_i$, so (by induction) there is a tree T' on $\{v_1, \ldots, v_n\} \setminus v_i$ with S' as its Prüfer code
  - Form T from T' by adding vertex v_i, joined only to σ_1
  - Prüfer code of T starts σ_1 and (by FACT 1) continues with S', so is S

Every string is the Prüfer code of some tree

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given string $S = (\sigma_1, \ldots, \sigma_{n-2})$ on alphabet $\{v_1, \ldots, v_n\}$
  - Find the lowest labeled vertex that does not appear in the string, v_i say
  - Form $S' = (\sigma_2, \ldots, \sigma_{n-2})$ by deleting first entry from S
  - S' is a string of length $n - 3$ on alphabet $\{v_1, \ldots, v_n\} \setminus v_i$, so (by induction) there is a tree T' on $\{v_1, \ldots, v_n\} \setminus v_i$ with S' as its Prüfer code
  - Form T from T' by adding vertex v_i, joined only to σ_1
  - Prüfer code of T starts σ_1 and (by FACT 1) continues with S', so is S

Map from Trees to Prüfer Codes is *Surjective*, so *BIJECTIVE*

Every string is the Prüfer code of some tree

Proof by Induction

- Case $n = 3$ easily verified
- For $n \geq 4$: given string $S = (\sigma_1, \ldots, \sigma_{n-2})$ on alphabet $\{v_1, \ldots, v_n\}$
  - Find the lowest labeled vertex that does not appear in the string, v_i say
  - Form $S' = (\sigma_2, \ldots, \sigma_{n-2})$ by deleting first entry from S
  - S' is a string of length $n - 3$ on alphabet $\{v_1, \ldots, v_n\} \setminus v_i$, so (by induction) there is a tree T' on $\{v_1, \ldots, v_n\} \setminus v_i$ with S' as its Prüfer code
  - Form T from T' by adding vertex v_i, joined only to σ_1
  - Prüfer code of T starts σ_1 and (by FACT 1) continues with S', so is S

Map from Trees to Prüfer Codes is *Surjective*, so *BIJECTIVE*

Cayley's Formula:

There are exactly n^{n-2} labelled trees on n vertices

The tree of a Prüfer code

Unraveling the Induction

- Given: a string S of length $n-2$ on alphabet $\{v_1, \ldots, v_n\}$, with $v_1 < v_2 \ldots < v_n$
- Repeat until S is empty and alphabet has size 2:
  - Identify the lowest letter in the alphabet that does not appear in the string, v_i say, and the first element of the string, v_j say
  - Add v_i to the graph being constructed (if it isn't already there), and join it to v_j (adding v_j to the graph first if necessary)
  - Remove v_i from the alphabet, and remove the first term from the string
- Join the two remaining vertices in the alphabet
- Result is a tree on vertex set $\{v_1, \ldots, v_n\}$ with Prüfer code S