Guided discovery in a discrete-mathematics course for math majors

Mary E. Flahive and John W. Lee
Mathematics Department, Oregon State University

MAA, PNW section
April 14, 2007

This work receives support from an NSF–DUE grant.

GOAL OF THE PROJECT:

To adapt Kenneth P. Bogart's successful "Teaching Introductory Combinatorics by Guided Group Discovery" for a required discrete mathematics class at Oregon State University.

OTHERS INVOLVED:

- Ken Bogart, Dartmouth College
- Barbara Edwards, Oregon State
- Rosa Orellana, Dartmouth College

OREGON STATE'S MATH MAJOR:

- First two years: calculus sequence, a course in applied differential equations, and a course in matrix algebra
- DM in first term of third year, concurrent with advanced calculus
- Third-year cohort

We began using Ken's notes in Fall 2003.

KEN BOGART'S OBJECTIVE: To design a course and a method in which "a large majority of the students would learn a large majority of the material of beginning combinatorics".

INFORMED BY:

- Davidson on discovery in small groups
- Schoenfeld on problem solving
- Dubinsky (et.al.) on genetic decomposition of mathematical knowledge

GUIDED GROUP DISCOVERY:

- Notes are a collection of interconnected problem sequences
- Students work together in groups during class
- Instructor is actively involved as a guide and collaborator in student learning

Our talk will concentrate on the notes.

GUIDED GROUP DISCOVERY:

- Notes are a collection of interconnected problem sequences
- Students work together in groups during class
- Instructor is actively involved as a guide and collaborator in student learning

Our talk will concentrate on the notes.

WHY ADAPT?
Differences between OSU and the Bogart sites:

- Larger class size Expansion of interstitial discussion and more summarizing material
- Less mathematical experience Our students have more trouble ferreting out definitions and results
- Less mathematical background Material from the original appendices needed
- Required course
  - Average student less highly motivated
  - Improvement in general mathematical maturity emphasized as much as the specific material
- DM rather than combinatorics For instance, more emphasis on induction and on equivalence relations.

WHY ADAPT?

Differences between OSU and the Bogart sites:

- Larger class size Expansion of interstitial discussion and more summarizing material
- Less mathematical experience Our students have more trouble ferreting out definitions and results
- Less mathematical background Material from the original appendices needed
- Required course
  - Average student less highly motivated
  - Improvement in general mathematical maturity emphasized as much as the specific material
- DM rather than combinatorics For instance, more emphasis on induction and on equivalence relations.

WHY ADAPT?

Differences between OSU and the Bogart sites:

- Larger class size Expansion of interstitial discussion and more summarizing material
- Less mathematical experience Our students have more trouble ferreting out definitions and results
- Less mathematical background Material from the original appendices needed
- Required course
  - Average student less highly motivated
  - Improvement in general mathematical maturity emphasized as much as the specific material
- DM rather than combinatorics For instance, more emphasis on induction and on equivalence relations.

WHY ADAPT?

Differences between OSU and the Bogart sites:

- Larger class size Expansion of interstitial discussion and more summarizing material
- Less mathematical experience Our students have more trouble ferreting out definitions and results
- Less mathematical background Material from the original appendices needed
- Required course
  - Average student less highly motivated
  - Improvement in general mathematical maturity emphasized as much as the specific material
- DM rather than combinatorics For instance, more emphasis on induction and on equivalence relations.

WHY ADAPT?
Differences between OSU and the Bogart sites:

- Larger class size Expansion of interstitial discussion and more summarizing material
- Less mathematical experience Our students have more trouble ferreting out definitions and results
- Less mathematical background Material from the original appendices needed
- Required course
  - Average student less highly motivated
  - Improvement in general mathematical maturity emphasized as much as the specific material
- DM rather than combinatorics For instance, more emphasis on induction and on equivalence relations.

WHY ADAPT?

Differences between OSU and the Bogart sites:

- Larger class size Expansion of interstitial discussion and more summarizing material
- Less mathematical experience Our students have more trouble ferreting out definitions and results
- Less mathematical background Material from the original appendices needed
- Required course
  - Average student less highly motivated
  - Improvement in general mathematical maturity emphasized as much as the specific material
- DM rather than combinatorics For instance, more emphasis on induction and on equivalence relations.

SOME DETAILS ON FUNCTION

Establishing a bijection is a fundamental combinatorial technique, and requires an evolution in the problem solver's understanding of function:

1. (Action Stage) Can plug values into a given function.
2. (Process Stage) Can determine whether or not a given relation is a function.
3. (Object Stage) Can reason about functions. In particular, is able to recognize how the notion of function is useful in a problem.
4. (Schema Stage) Familiarity with a number of examples, processes, and objects related to function, and has formed mental links among them.

In the original notes, Ken assumed all entering students would have a process conception of function and would have done exercises building toward an object stage.

SOME DETAILS ON FUNCTION

Establishing a bijection is a fundamental combinatorial technique, and requires an evolution in the problem solver's understanding of function:

1. (Action Stage) Can plug values into a given function.
2. (Process Stage) Can determine whether or not a given relation is a function.
3. (Object Stage) Can reason about functions. In particular, is able to recognize how the notion of function is useful in a problem.
4. (Schema Stage) Familiarity with a number of examples, processes, and objects related to function, and has formed mental links among them.

In the original notes, Ken assumed all entering students would have a process conception of function and would have done exercises building toward an object stage.

SOME DETAILS ON FUNCTION

Establishing a bijection is a fundamental combinatorial technique, and requires an evolution in the problem solver's understanding of function:

1. (Action Stage) Can plug values into a given function.
2. (Process Stage) Can determine whether or not a given relation is a function.
3. (Object Stage) Can reason about functions. In particular, is able to recognize how the notion of function is useful in a problem.
4. (Schema Stage) Familiarity with a number of examples, processes, and objects related to function, and has formed mental links among them.

In the original notes, Ken assumed all entering students would have a process conception of function and would have done exercises building toward an object stage.

SOME DETAILS ON FUNCTION

Establishing a bijection is a fundamental combinatorial technique, and requires an evolution in the problem solver's understanding of function:

1. (Action Stage) Can plug values into a given function.
2. (Process Stage) Can determine whether or not a given relation is a function.
3. (Object Stage) Can reason about functions. In particular, is able to recognize how the notion of function is useful in a problem.
4. (Schema Stage) Familiarity with a number of examples, processes, and objects related to function, and has formed mental links among them.

In the original notes, Ken assumed all entering students would have a process conception of function and would have done exercises building toward an object stage.

SOME DETAILS ON FUNCTION

Establishing a bijection is a fundamental combinatorial technique, and requires an evolution in the problem solver's understanding of function:

1. (Action Stage) Can plug values into a given function.
2. (Process Stage) Can determine whether or not a given relation is a function.
3. (Object Stage) Can reason about functions. In particular, is able to recognize how the notion of function is useful in a problem.
4. (Schema Stage) Familiarity with a number of examples, processes, and objects related to function, and has formed mental links among them.

In the original notes, Ken assumed all entering students would have a process conception of function and would have done exercises building toward an object stage.

SOME DETAILS ON FUNCTION

Establishing a bijection is a fundamental combinatorial technique, and requires an evolution in the problem solver's understanding of function:

1. (Action Stage) Can plug values into a given function.
2. (Process Stage) Can determine whether or not a given relation is a function.
3. (Object Stage) Can reason about functions. In particular, is able to recognize how the notion of function is useful in a problem.
4. (Schema Stage) Familiarity with a number of examples, processes, and objects related to function, and has formed mental links among them.

In the original notes, Ken assumed all entering students would have a process conception of function and would have done exercises building toward an object stage.

AN EXAMPLE OF A PROBLEM-SEQUENCE ON BIJECTION

(From an actual class experience at OSU in which the class developed a game which reinforced the use of the Bijection Principle.)

Next you will explore the idea of *labelled* trees. Figure A gives all different labellings of a fixed tree with 3 vertices. Notice that the convention for labelling the vertices of trees is that the tree which has edges between vertices 1 and 2 and between vertices 2 and 3 is different from the tree that has edges between vertices 1 and 3 and between vertices 2 and 3.

In notes, Figure A is here.

AN EXAMPLE OF A PROBLEM-SEQUENCE ON BIJECTION

(From an actual class experience at OSU in which the class developed a game which reinforced the use of the Bijection Principle.)

Next you will explore the idea of *labelled* trees. Figure A gives all different labellings of a fixed tree with 3 vertices. Notice that the convention for labelling the vertices of trees is that the tree which has edges between vertices 1 and 2 and between vertices 2 and 3 is different from the tree that has edges between vertices 1 and 3 and between vertices 2 and 3.

In notes, Figure A is here.

Some experimentation leading to a conjecture

Problem: How many labelled trees are there on the vertex set [2]? On the vertex set [3]? How many labelled trees are there on four vertices? How many labelled trees are there with five vertices? You don't have a lot of data to formulate a guess, but try to guess a formula for the number of labelled trees with vertex set $[n]$. When you get to four and especially five vertices, draw all the unlabelled trees you can think of, and then figure out in how many different ways you can put labels on the vertices.

Designing an auxiliary sequence

The next problems will develop a method for proving the formula you just guessed in the last problem. In order to do this, an auxiliary sequence is defined.

In the notes, the notion of **Prüfer code** associated with a labelled tree is now introduced. The code is a sequence $b_1, b_2 \ldots$ of positive integers which is defined inductively. We omit the exact definition here.

Calculation of some P-codes.

Problem: Use the letter B to stand for the sequence of b_is inductively obtained in this way. For the tree in Figure A, the sequence B is 2344378. At this point, work with your group to draw some other labelled trees on eight vertices and construct the sequence B associated with each tree.

Developing some basic properties of P-codes

Problem: Use your earlier work to answer the questions posed in the above two-step algorithm.

Problem: How long is the sequence B computed from a labelled tree with n vertices? Explain.

Problem: From your examples, decide if you can predict the last member of the sequence B. Explain.

Problem: Is it possible for a_1 to be in B? Can you tell from B what a_1 is?

Setting up the function.

For a labelled tree T, the associated sequence $P(T) := b_1, b_2, \ldots, b_{n-2}$ is called a **Prüfer code** of T. For instance, the Prüfer code for the labelled tree T of Figure A is $P(T) = 234437$. Notice that the last term of B is not included in the Prüfer code because it's known to be n.

They next look at the special case of $n = 9$:

Problem: Let $\mathcal{S}_9$ be the set of all labelled trees on nine vertices. For each tree $T \in \mathcal{S}_9$, define $P(T)$ to be the Prüfer code for T.

Problem: Why is the relation $\{(T, P(T)) \ : \ T \in \mathcal{S}_9\}$ a function with domain $\mathcal{S}_9$? Find a co-domain for this function. (At this point I'm not asking you to find the *smallest* co-domain.)

Setting up the function.

For a labelled tree T, the associated sequence $P(T) := b_1, b_2, \ldots, b_{n-2}$ is called a **Prüfer code** of T. For instance, the Prüfer code for the labelled tree T of Figure A is $P(T) = 234437$. Notice that the last term of B is not included in the Prüfer code because it's known to be n.

They next look at the special case of $n = 9$:

Problem: Let $\mathcal{S}_9$ be the set of all labelled trees on nine vertices. For each tree $T \in \mathcal{S}_9$, define $P(T)$ to be the Prüfer code for T.

Problem: Why is the relation $\{(T, P(T)) : T \in \mathcal{S}_9\}$ a function with domain $\mathcal{S}_9$? Find a co-domain for this function. (At this point I'm not asking you to find the *smallest* co-domain.)

Game designed by the Fall 2003 class.

Problem: Play the following game in your group: Each of you should individually choose a $T \in \mathcal{S}_9$ and secretly find its Prüfer code, $P(T)$. The other members of the group should try to find a labelled graph which has your sequence as its code. Is there only one? What does your answer say about the function P?

Problem: Now, as a group write down any sequence of seven integers from [9]. Try to find a tree $T \in \mathcal{S}_9$ for which your sequence is $P(T)$. Do this for several different sequences. Use this information to find the smallest co-domain for the function P.

Returning to the general case.

Problem: Find a bijection between the set of labelled trees with n vertices and another set that you already know how to count.
Problem: Find the number of labelled trees with n vertices. Is this the formula you conjectured earlier ?

EVOLUTION OF THE BIJECTION IN FALL 2003

- At the start of class, most of the groups were beginning Ken's problem sequence on the enumeration of labeled trees.
- By the end of the first hour, many groups saw a pattern emerging from counting the number for small vertex sets. Some students were ready to accept the conjecture without proof, while others wanted to find a general argument.
- After a short break, most students agreed to forge ahead and work on understanding Prüfer codes.
- One group thought of the idea of a game, and soon all were playing it.
- By the end of class, all groups had played the game and most students were convinced there would always exist a unique tree for every Prüfer code.
- Many left class trying to establish the necessary bijection by describing the sequences which were guaranteed to be Prüfer codes.

EVOLUTION OF THE BIJECTION IN FALL 2003

- At the start of class, most of the groups were beginning Ken's problem sequence on the enumeration of labeled trees.
- By the end of the first hour, many groups saw a pattern emerging from counting the number for small vertex sets. Some students were ready to accept the conjecture without proof, while others wanted to find a general argument.
- After a short break, most students agreed to forge ahead and work on understanding Prüfer codes.
- One group thought of the idea of a game, and soon all were playing it.
- By the end of class, all groups had played the game and most students were convinced there would always exist a unique tree for every Prüfer code.
- Many left class trying to establish the necessary bijection by describing the sequences which were guaranteed to be Prüfer codes.

EVOLUTION OF THE BIJECTION IN FALL 2003

- At the start of class, most of the groups were beginning Ken's problem sequence on the enumeration of labeled trees.
- By the end of the first hour, many groups saw a pattern emerging from counting the number for small vertex sets. Some students were ready to accept the conjecture without proof, while others wanted to find a general argument.
- After a short break, most students agreed to forge ahead and work on understanding Prüfer codes.
- One group thought of the idea of a game, and soon all were playing it.
- By the end of class, all groups had played the game and most students were convinced there would always exist a unique tree for every Prüfer code.
- Many left class trying to establish the necessary bijection by describing the sequences which were guaranteed to be Prüfer codes.

EVOLUTION OF THE BIJECTION IN FALL 2003

- At the start of class, most of the groups were beginning Ken's problem sequence on the enumeration of labeled trees.
- By the end of the first hour, many groups saw a pattern emerging from counting the number for small vertex sets. Some students were ready to accept the conjecture without proof, while others wanted to find a general argument.
- After a short break, most students agreed to forge ahead and work on understanding Prüfer codes.
- One group thought of the idea of a game, and soon all were playing it.
- By the end of class, all groups had played the game and most students were convinced there would always exist a unique tree for every Prüfer code.
- Many left class trying to establish the necessary bijection by describing the sequences which were guaranteed to be Prüfer codes.

EVOLUTION OF THE BIJECTION IN FALL 2003

- At the start of class, most of the groups were beginning Ken's problem sequence on the enumeration of labeled trees.
- By the end of the first hour, many groups saw a pattern emerging from counting the number for small vertex sets. Some students were ready to accept the conjecture without proof, while others wanted to find a general argument.
- After a short break, most students agreed to forge ahead and work on understanding Prüfer codes.
- One group thought of the idea of a game, and soon all were playing it.
- By the end of class, all groups had played the game and most students were convinced there would always exist a unique tree for every Prüfer code.
- Many left class trying to establish the necessary bijection by describing the sequences which were guaranteed to be Prüfer codes.

EVOLUTION OF THE BIJECTION IN FALL 2003

- At the start of class, most of the groups were beginning Ken's problem sequence on the enumeration of labeled trees.
- By the end of the first hour, many groups saw a pattern emerging from counting the number for small vertex sets. Some students were ready to accept the conjecture without proof, while others wanted to find a general argument.
- After a short break, most students agreed to forge ahead and work on understanding Prüfer codes.
- One group thought of the idea of a game, and soon all were playing it.
- By the end of class, all groups had played the game and most students were convinced there would always exist a unique tree for every Prüfer code.
- Many left class trying to establish the necessary bijection by describing the sequences which were guaranteed to be Prüfer codes.