Class 39: Association Lists and Searching

Held: Tuesday, 7 November 2006

Summary: Today we visit a database-like application of Scheme, Association lists. Association lists allow you to associate information with a “key” and then look up the value by the key.

Related Pages:
- EBoard.
- Lab: Association Lists.
- Reading: Association Lists.

Due
- HW12.

Notes:
- Don’t forget Thursday’s Thursday Extra on the GIMP. Noon. Free Pizza.

Overview:
- Databases and dictionaries.
- Lists as dictionaries.
- assoc, a procedure for searching lists.
- Variants of Association Lists.
- Lab.

Simple Database Problems
- Databases are among the most common applications of computers.
  - After all, there’s a reason that Larry Ellison is nearly as rich as Bill Gates.
- A database is a mechanism for storing data so that you can easily access the data you need.
- One simple database activity is looking up values by keys.
- Databases that provide only that activity are called dictionaries.

Association Lists
- In Scheme, dictionaries are typically implemented with a data structure known as the association list.
- An association list is a list of elements each of which has a key as its car.
- You can use the (assoc key list) procedure to look up values by key.
Searching in Association Lists

- Suppose Scheme didn’t include `assoc`. How would you write it? Probably recursively.
- If the list is empty, it does not contain the value.
- If the key of the first element in the list is the key we’re looking for, return the corresponding value.
- Otherwise, look in the rest of the list.
- This technique is called sequential search.

Variants of Association Lists

- Given that we can write our own `assoc` procedure, we can easily implement a number of interesting variants of association lists.
- For example, if the same key appears multiple times in the association list, we might return all matching values (rather than the first matching value).
- Similarly, instead of searching by key, we might search by predicate.

alists-lab

- Do the lab.