AM [Lenat, 82]

AM stands for “Automated Mathematician”

Starts with “prenumerical concepts” and discovers concepts in number theory.

System *learns by exploration:* Search without a fixed goal.

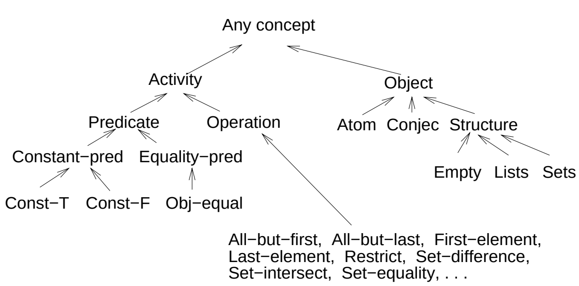
Note: Never *proves* anything

AM models *mathematical discovery*.

In particular, it

- Starts with about 100 concepts
- Hypothesizes relationships between them
- Defines new concepts to investigate

Part of AM’s initial concept network



Concept representation:

- Concepts are described in frames with slots for definitions, specializations, generalizations, examples, relevant conjectures . . .
- Multiple definitions are possible, but one form is Lisp code.

AM's concept for prime numbers

Name: Prime-numbers

Definitions:

Origin: Number-of-divisors-of(x) = 2

Iterative: (for X $>$ 1): For i = 2, $\sqrt{x}$, (not i | x)

Examples: 2, 3, 5, 7, 11, 13, 17

boundary: 2, 3

boundary-failures: 0, 1

failures: 12

Gens: Numbers, numbers w/even number of divisors

Specs: Odd primes

Conjectures: Unique fact., Goldbach's conjecture

Interest: Relation with times, with divisors-of

Worth: 800

Some heuristics used by AM

- If f is a function from A to B, and B is ordered, then consider the elements of A that are mapped into extremal elements of B. Create a new concept representing this subset of A.
- If some (but not most) examples of some concept X are also examples of another concept Y, create a new concept representing the intersection of X and Y.
- If very few examples of a concept X are found, then add to the agenda the task of finding a generalization of X.

A more detailed example

Heuristic for generalizing predicates that answer "true" very rarely:

If	task was "Fill in examples of X"
	and over 100 items are known in domain(X)
	and task has executed at least 10 seconds
	and X returned T at least once
	and ratio of F to T is at least 20
Then	add the task: "Fill in generalizations of X"

Example:

- Task is "fill in examples of list equality."
- Heuristic for the task: Try random examples.
- Result: Seldom true.
- Previous heuristic suggests a new task: Generalize list-equality.

New task turns out to be a good choice . . .

How AM discovers numbers

- Starts out knowing about $list$, set, and bag, and equality for them.
- Because few random lists are equal to each other, decides to weaken the definition of list equality.

Weakening list equality

Initial definition of **list-equal**:

```
To test if list L1 is equal to list L2
  If they are both empty,
  Then yes.
  Else if they are both nonempty,
       Then if their first elements are equal,
            Then check recursively if the c-
drs of
              the lists are list-equal.
            Else no.
       Else no.
```


One way to weaken a test is to drop a condition.

This gives **gen-equal**:

```
To test if list L1 is gen-equal to list L2
  If they are both empty,
  Then yes.
  Else if they are both nonempty,
       Then if True
            Then check recursively if the c-
drs or
              the lists are gen-equal.
            Else no.
       Else no.
```

Gen-equal is true more often, so it's more interesting . . .

- AM applies the heuristic: "If weakening a relation gives an interesting concept, look for a form for objects so that the old relation gives the same answer as the weaker one."
- It knows how to make the heads of a list equal: make them both t.
- This gives rise to lists of the form (t t t t . . .).
- These correspond to numbers—concatenation of these lists is addition!

How AM discovers prime numbers

- Finds the natural numbers, addition, multiplication, and their inverses.
- Creates concept of "divisibility," and notices some numbers have few divisors.
- One of its heuristics is to explore extreme cases. Applies it to numbers with few divisors:
  - Looks for numbers with no divisors. Fails.
  - Looks for numbers with one divisor. Finds only "1".
  - Looks for numbers with two divisors. (Lenat names them: "prime numbers.")

- Continues to conjecture various results:
  - Unique factorization.
  - Goldbach's conjecture (All even numbers greater than 2 can be expressed as the sum of two primes).

Result is the previous concept for prime numbers:

Name: Prime-numbers
Definitions:
  Origin: Number-of-divisors-of(x) = 2
  Iterative: (for X $>$ 1): For i = 2, $\sqrt{x}$, (not i $|$ x)
Examples: 2, 3, 5, 7, 11, 13, 17
  boundary: 2, 3
  boundary-failures: 0, 1
  failures: 12
Gens: Numbers, numbers w/even number of divisors
Specs: Odd primes
Conjectures: Unique fact., Goldbach's conjecture
Interest: Relation with times, with divisors-of
Worth: 800

Relationship to other methods

- Generates its own target concepts (unlike version spaces, ID3, IBL, EBL). More flexible than FDL.
- Uses knowledge, but not just knowledge transformation (unlike EBL)

Lessons from AM

- Since it first did so well, why didn't it continue?

  Static heuristics; exhaustion of initial knowledge

- Why didn't it work as well in other domains?

  Relied on

  - Lucky connection between syntax and semantics
  - Clean domain and simplification of search process (no planning was needed)

Lenat's conclusion: Interesting learning requires knowledge. Lenat moved to CYC . . .