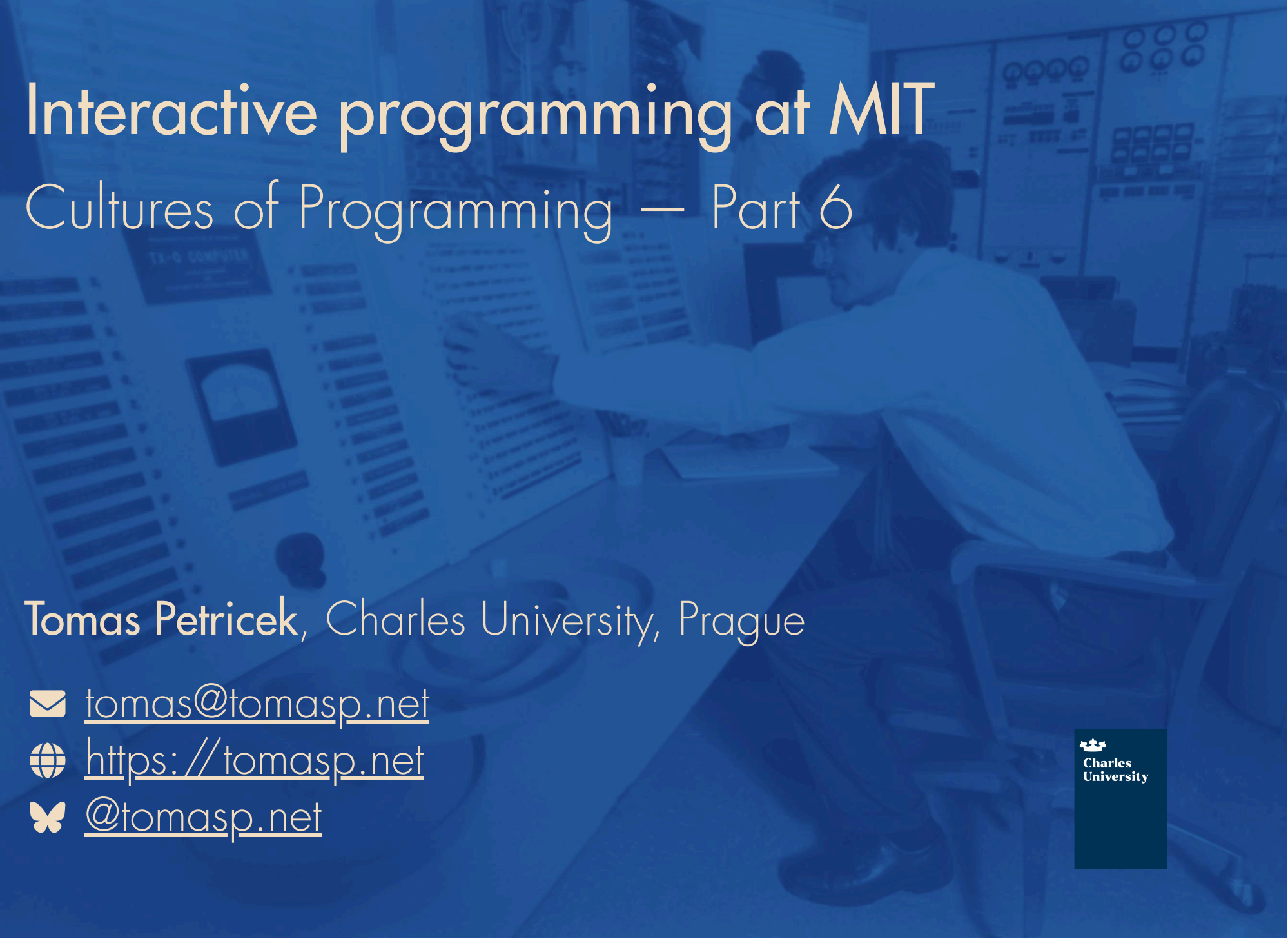


Interactive programming at MIT

Cultures of Programming — Part 6

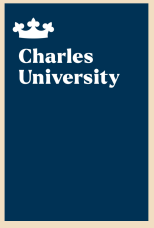
Tomas Petricek, Charles University, Prague

tomas@tomasp.net
https://tomasp.net
@tomasp.net

Charles University

Spacewar!

TX-0 and PDP-1 computers



Computing in the late 1950s

Batch processing

- Hand punched cards to operator
- Get "Expects a comma. Abort"



TX-0 (1956) and TX-2 (1958)

- Test transistor technology
- Donated to MIT RLE, available for interactive use

DEC PDP-1 (1959)

- More powerful TX-inspired commercial computer
- Available at BBN and donated to MIT

Charles University



Spacewar!

Steve "Slug" Russell & Tech Model Railroad Club (TMRC) members

Sine and cosine story… "What's your excuse?"

Demo showcasing PDP-1 capabilities

Charles University

Spacewar! and hacker culture

Fun and late-night hacking



- Lowest priority thing to do!
- Less demand during the nights

Technical challenges

- Starmap with 1000 real stars
- Generating instructions to draw rotated spaceship

Information sharing culture

- Improvements to code encouraged!
- Practical hands-on knowledge

Charles University

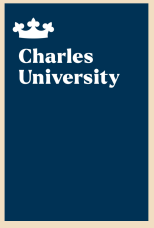
Interactive computers

- **Whirlwind computer for SAGE** (1951)
  Cathode-ray-tube (CRT) screen and light pen input
- **Transistorized experimental computer (TX-0)**
  Interactive use, toggle switch storage, core memory
- **Transistorized experimental computer (TX-2)**
  64k of 36bit word memory, graphics and light pen
- **LINC "personal computer"** (1962)
  Real-time processing, medical applications

Charles University

Interactive programming

TX-0 and PDP-1 computers





On-line debugging

"For the past few years it has been regarded as a great professional sin, but like most sins, it still exists, is rarely talked about, and hardly ever admitted."

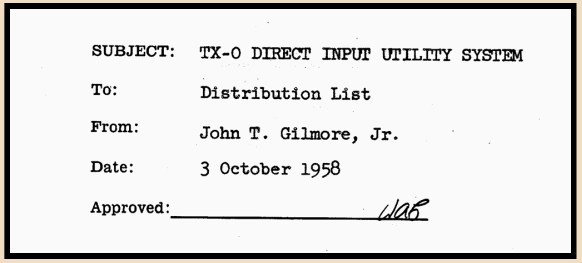
Technical innovations

- Toggle switch storage
- Flexowriter terminal
- CRT screen

TX-0 Direct Input

Utility Tape 3 (UT-3)

- Accept commands from Flexotype
- Modify memory through switch storage
- Search memory, edit, run, print diff



SUBJECT: TX-0 DIRECT INPUT UTILITY SYSTEM

To: Distribution List

From: John T. Gilmore, Jr.

Date: 3 October 1958

Approved: _____ JTG

TX-0 Direct Input Utility System

- Try "some new programming techniques"
- Conversion of symbolic in-memory program
- Provide "immediate information" to help find errors

Charles University

Interactive programming

TX-0 Direct Input Utility System features

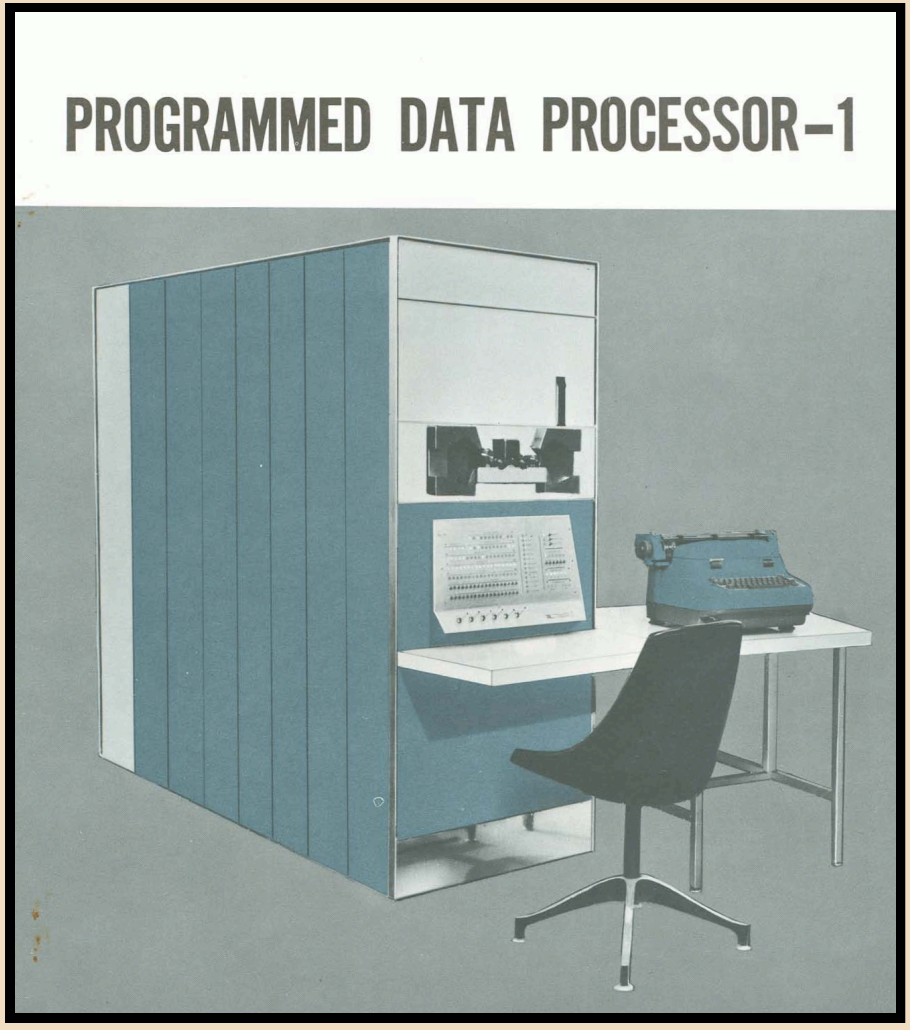
 **Hark** - modify in-memory instruction and dump diff

 **Find** - search memory for a value (20 seconds)

 **Outlaw** - how did control get to a given point

 **Flow Chart Display** - visually trace program execution

Charles University



Costs of direct use

PDP-1 costs $120,000 in 1960 (~1.2M now)

Interactive use too expensive in practice!

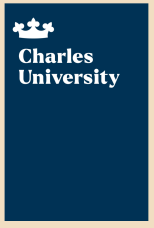
Reflected by nicknames

- Expensive Synthesiser
- Expensive Typewriter
- Expensive Planetarium

Charles University

Time-sharing

Making interactive use practical



Time-sharing

A Time Sharing Operator Program

 Let multiple users interact with a single machine

 Most of the time spent thinking and typing!

 Idea "seemed quite inevitable" (McCarthy)

 Thought "everybody had in mind to do" it

Charles University



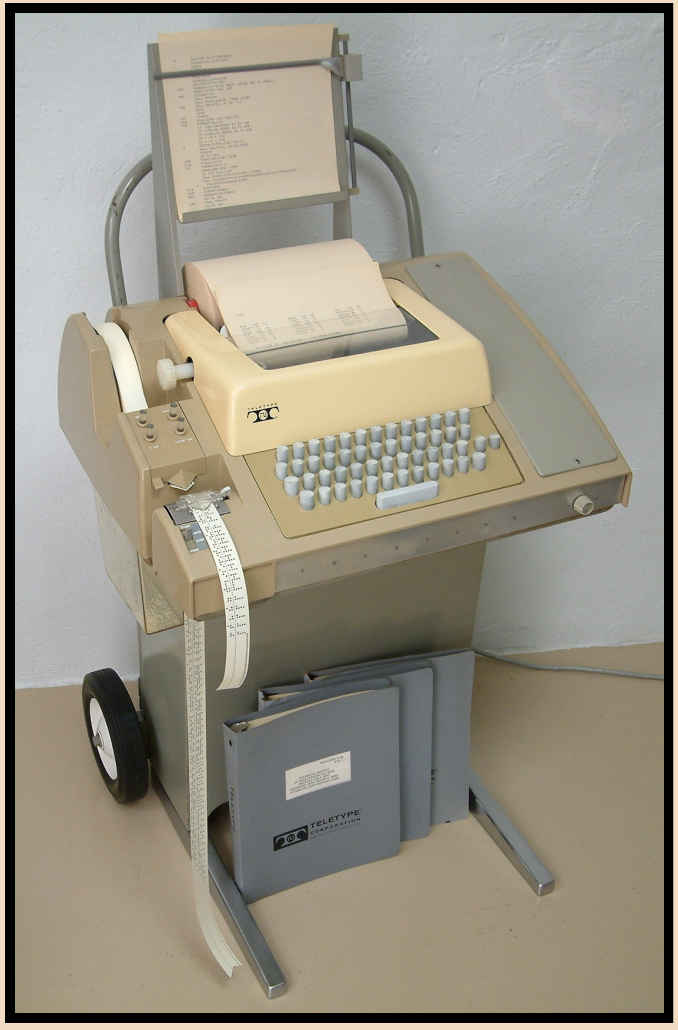
Major undertaking, but eventually...

Compatible Time-Sharing System (CTSS), funded by Project MAC (ARPA)

IBM 7094 (mid-1960s)

- Over 100 terminals
- 30 concurrent user

Charles University



Teletype terminal

Way more limited interaction than TX-0!

Slow line-based input and output

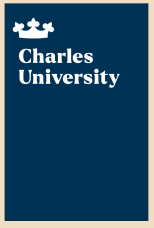
Initially no screen

Different kind of programming is needed...

Charles University

Lisp

As a language for hackers

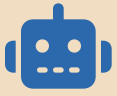


History of Lisp

Correcting misconceptions

≠ **Lisp is the first functional language**

Started as FORTRAN library & is imperative!

 **Lisp was based on the lambda calculus**

Motivated by needs of symbolic AI

</> **Code as data motivates S-expressions**

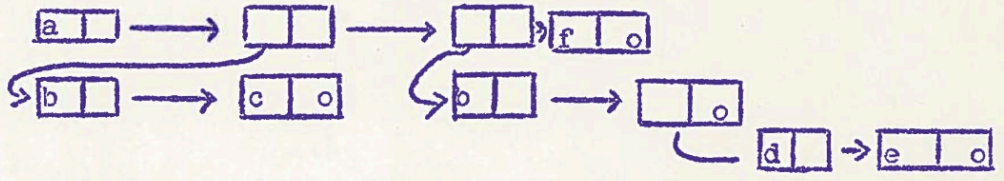
First versions had more ordinary M-expressions

An Algebraic Language for the Manipulation of Symbolic Expressions (McCarthy, 1958)

3. Representation of Formal Expressions by List Structures and the Basic List Structure Operations

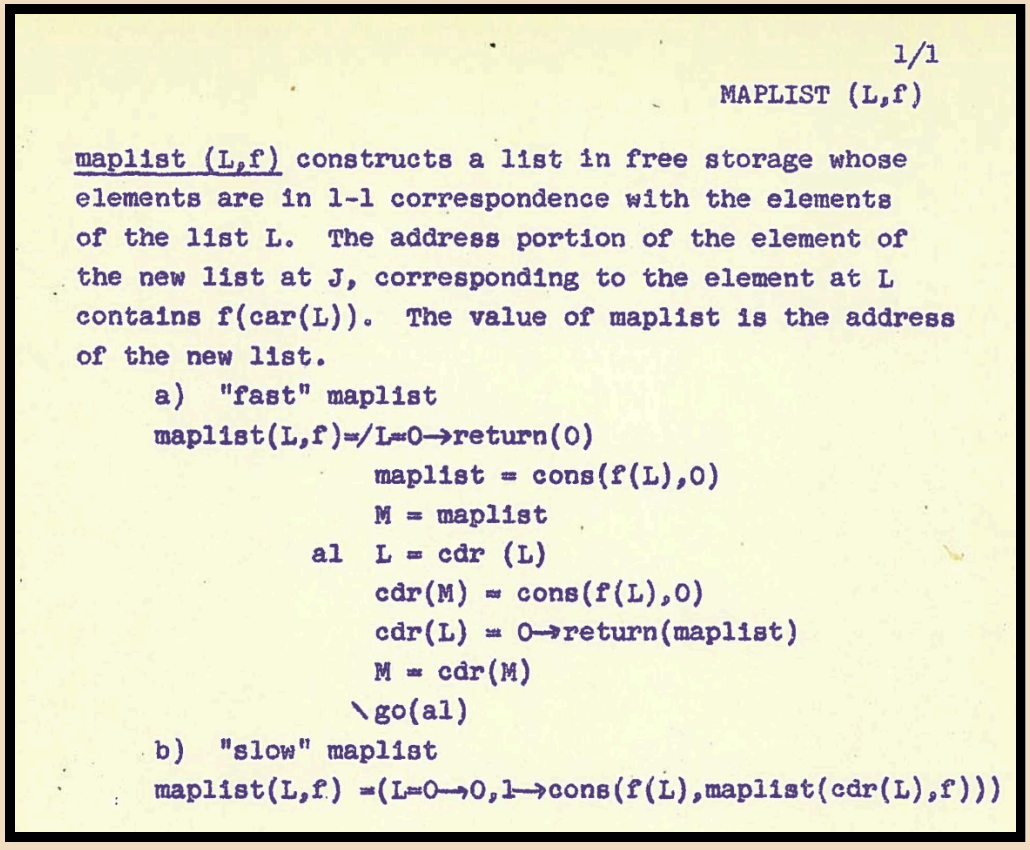
3.1. Representation of Expression.

The use of list structures for storing data corresponds to the mathematical system which uses the sequences as the basic expression. Thus, the sequence (a, (b,c), (b, (d,e)), f) corresponds to the list structure



Code as data becomes an appealing feature for AI!
Meta-programming (aka Recursive Self Improvement)

Charles University



1/1

MAPLIST (L,f)

maplist (L,f) constructs a list in free storage whose elements are in 1-1 correspondence with the elements of the list L. The address portion of the element of the new list at J, corresponding to the element at L contains f(car(L)). The value of maplist is the address of the new list.

a) "fast" maplist

```
maplist(L,f)=/L=0→return(0)
             maplist = cons(f(L),0)
             M = maplist
         a1  L = cdr (L)
             cdr(M) = cons(f(L),0)
             cdr(L) = 0→return(maplist)
             M = cdr(M)
            \go(a1)
```

b) "slow" maplist

```
maplist(L,f) =(L=0→0,1→cons(f(L),maplist(cdr(L),f)))
```

LISP 1 (1959)

- List processing
- Imperative features
- High-level language

LISP 1.5 (1961)

Programmer's Manual published in 1961

Charles University

Lisp hackers

The birth of the eval function

- Implementation by Steve "Slug" Russell
- Function `eval` written as mathematical model
- Russell points out this is an interpreter
- Lisp becomes "meta-circular" by accident!

Charles University

Towards interactive Lisp

Non-interactive Lisp

- Prefix program with 11 cards from "Utility Desk" drawer (room 26-265)



Testing time-sharing on IBM 704

- Evaluates entered expression
- "TEN-Mode" to make entry easier

Interactive Lisp for PDP-1

- L Peter Deutsch at BBN
- Needs mechanism for rapid program editing!

Charles University

```
(LAMBDA (X) Y
  (COND ((NULL X) Y)
  (T (CONS (CAR X)
    (APPEND (CDR X) Y)))))

*P
   (LAMBDA (X) Y (COND & &)).
*2 P
   (X)
*(3)
*(2 (X Y))
*P
   (LAMBDA (X Y) (COND & &)).
```

Interactive Lisp editor (1963)

Fixing wrong argument list `(X) Y` to `(X Y)`

`P` to print outline

`2 P` to focus and print

`(3)` to delete third item

`(2 (..))` to replace

Charles University

Interactive programming at MIT



Hacker culture dominant

Direct access to computers, liberal policies



Mostly documented in informal memos

Disregard for conventional academic values



Contrast to traditional business use

Using computer interactively is a waste of money!



Enabled by new computer interfaces

Direct use of TX-0 and time-sharing teletype

Charles University