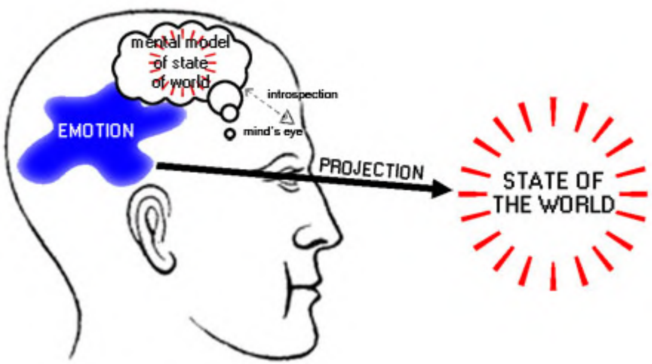
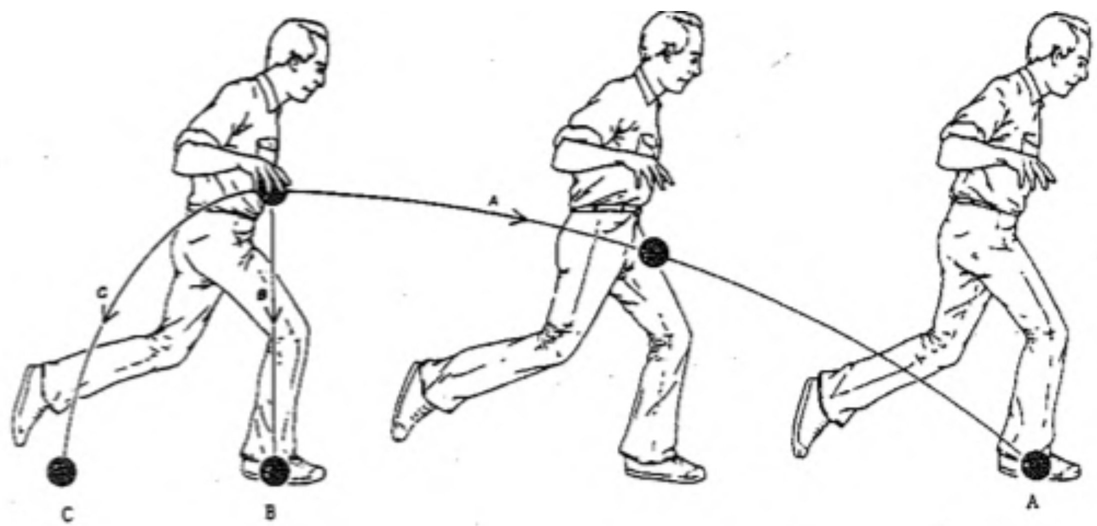
chapter 12

Cognitive Models

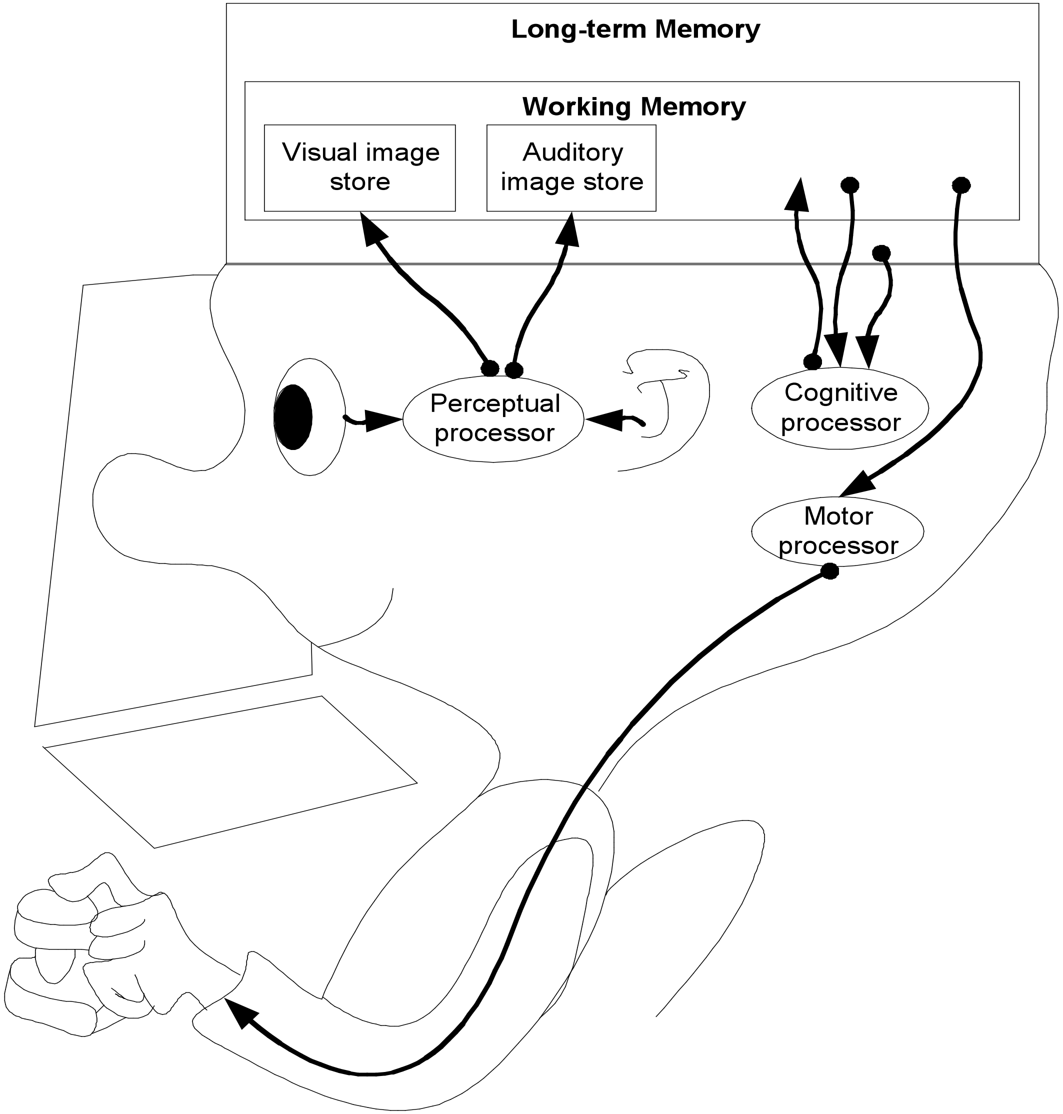
Humans create mental models to explain behavior





Cognitive Models

How do users **perceive**,
think
and **act**



Donald Norman's Model

- **Seven stages**
  1. User establishes the goal
  2. Formulates intention
  3. Specifies actions at interface
  4. Executes action
  5. Perceives system state
  6. Interprets system state
  7. Evaluates system state with respect to goal

- **Norman's model concentrates on user's view of the interface**

Donald Norman's Model

Some systems are harder to use than others

Why?

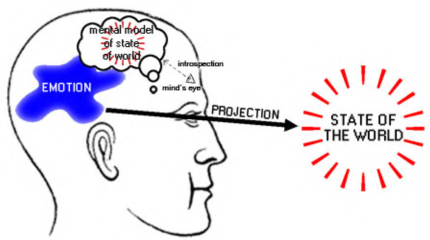
Gulf of Execution

user's formulation of actions ≠ actions allowed by the system

Gulf of Evaluation

user's expectation of changed system state ≠ actual presentation of this state

Cognitive Models



- **Hierarchical models**

  Represent a user’s task and goal structure

- **Linguistic models**

  Represent the user-system grammar

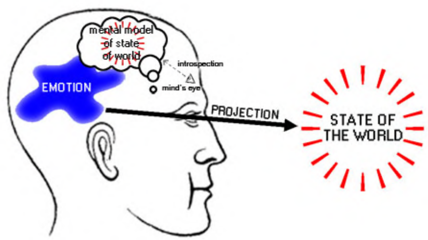
- **Physical and device models**

  Represent human motor skills

- **Cognitive architecture**

  Underline all all of these cognitive models

Cognitive Models



- **They model aspects of user:**
  - understanding
  - knowledge
  - intentions
  - processing

→ user behavior

- **Common categorization:**
  - Competence vs. Performance
  - Computational flavour
  - No clear divide

Hierarchical models

Goal and task hierarchies

- Mental processing as divide-and-conquer
- Example: to make a sales report we have to

produce report
gather data
.   **find** book names
.   .   **do** keywords search of names database
.   .   .   … *further sub-goals*
.   .   **sift** through names and abstracts by hand
.   .   .   … *further sub-goals*
.   **search** sales database - further sub-goals
layout tables and histograms => further sub-goals
write description => further sub-goals

The GOMS Cognitive Model

Goals
- What the user wants to achieve

Operators
- Basic actions user performs

Methods
- Decomposition of a goal into subgoals/operators

Selection
- Means of choosing between competing methods

GOMS example

```
GOAL: CLOSE-WINDOW
.   [select GOAL: USE-MENU-METHOD
            .   MOVE-MOUSE-TO-FILE-MENU
            .   PULL-DOWN-FILE-MENU
            .   CLICK-OVER-CLOSE-OPTION
            GOAL: USE-CTRL-W-METHOD
            .   PRESS-CONTROL-W-KEYS]
```

For a particular user (e.g. Sam):

Rule 1: Select USE-MENU-METHOD unless another rule applies

Rule 2: If the application is GAME, select CTRL-W-METHOD

Issues for goal hierarchies

- Granularity
  - Where do we start?
  - Where do we stop?
- Routine learned behavior, not problem solving
  - The unit task
- Conflict
  - More than one way to achieve a goal
- Error Detection (e.g. some British ATMs)

GOMS and Closure

```
GOAL: GET-MONEY
. GOAL: USE-ATM
. . INSERT-CARD
. . ENTER-PIN
. . ENTER-AMOUNT
. . COLLECT-MONEY
      << outer goal now satisfied goal stack popped >>
. . COLLECT-CARD – subgoal operators missed
```

What was the problem?

Closure achieved within a sub-goal, before complete all actions in that level.

A success case

DESIGN FOCUS

GOMS saves money

Some years ago the US telephone company NYNEX were intending to install a new computer system to support their operators. Before installation a detailed GOMS analysis was performed taking into account the cognitive and physical processes involved in dealing with a call. The particular technique was rather different from the original GOMS notation as described here. Because an operator performs several activities in parallel a PERT-style GOMS description was constructed [192, 154]. The PERT analysis was used to determine the critical path, and hence the time to complete a typical task. It was discovered that rather than speeding up operations, the new system would take longer to process each call. The new system was abandoned before installation, leading to a saving of many millions of dollars.

Exercise

Create a ***GOMS*** *description of the task of photocopying an article from a journal. Discuss the issue of closure in terms of your GOMS description.*

Exercise

```
GOAL: PHOTOCOPY-PAPER
.     GOAL: LOCATE-ARTICLE
.     GOAL: PHOTOCOPY-PAGE repeat until no more pages
.     .     GOAL: ORIENT-PAGE
.     .     .   OPEN-COVER
.     .     .   SELECT-PAGE
.     .     .   POSITION-PAGE
.     .     .   CLOSE-COVER
.     .     GOAL: PRESS-COPY-BUTTON
.     .     GOAL: VERIFY-COPY
.     .     .   LOCATE-OUT-TRAY
.     .     .   EXAMINE-COPY
.     GOAL: COLLECT-COPY
.     .     LOCATE-OUT-TRAY
.     .     REMOVE-COPY
.     GOAL: RETRIEVE-JOURNAL
.     .     OPEN-COVER
.     .     REMOVE-JOURNAL
.     .     CLOSE-COVER
```

Exercise

```
GOAL: PHOTOCOPY-PAPER
.     GOAL: LOCATE-ARTICLE
.     GOAL: PHOTOCOPY-PAGE repeat until no more pages
.     .     GOAL: ORIENT-PAGE
.     .     .   OPEN-COVER
.     .     .   SELECT-PAGE
.     .     .   POSITION-PAGE
.     .     .   CLOSE-COVER
.     .     GOAL: PRESS-COPY-BUTTON
.     .     GOAL: VERIFY-COPY
.     .     .   LOCATE-OUT-TRAY
.     .     .   EXAMINE-COPY
.     GOAL: COLLECT-COPY
.     .     LOCATE-OUT-TRAY
.     .     REMOVE-COPY (outer goal satisfied => error)
.     GOAL: RETRIEVE-JOURNAL
.     .     OPEN-COVER
.     .     REMOVE-JOURNAL
.     .     CLOSE-COVER
```

Linguistic models

Linguistic Notations

- Understanding the **user's behavior** and **cognitive difficulty** based on analysis of language between user and system.

- Similar in emphasis to dialogue models

- Backus–Naur Form (**BNF**)
- Task–Action Grammar (**TAG**)

Backus-Naur Form (BNF)

- Very common notation from computer science
- A purely syntactic view of the dialogue, here from the user perspective
- **Terminals**
  - lowest level of user behavior
  - e.g. CLICK-MOUSE, MOVE-MOUSE
- **Nonterminals**
  - ordering of terminals
  - higher level of abstraction
  - e.g. select-menu, position-mouse

Example of BNF

- **Basic syntax**:
  - nonterminal ::= expression
- **An expression**
  - contains terminals and nonterminals
  - combined in sequence (+) or as alternatives (|)

```
draw line     ::=  select line + choose points + last point
select line   ::=  pos mouse + CLICK MOUSE
choose points::=  choose one | choose one + choose points
choose one    ::=  pos mouse + CLICK MOUSE
last point    ::=  pos mouse + DBL CLICK MOUSE
pos mouse     ::=  NULL | MOVE MOUSE + pos mouse
```

Difficulty Measurements with BNF

- **Number of** rules (not so good)
- **Number of** + and | operators
- **Complications**
  - Same syntax for different semantics
  - No reflection of user's perception of system response
  - Minimal consistency checking (up ≠ down)

Task Action Grammar (TAG)

- Making consistency more explicit
- Encoding user's world knowledge
- Parameterized grammar rules
- Nonterminals are modified to include additional semantic features

Consistency in TAG

- In BNF, three UNIX commands would be described as:

```
copy::= cp+filename+filename | cp+filenames+directory
move::= mv+filename+filename | mv+filenames+directory
link::= ln+filename+filename | ln+filenames+directory
```

- No BNF measure could distinguish between this and a less consistent grammar in which

```
link::= ln+filename+filename | ln+directory+filenames
```

Consistency in TAG (cont'd)

- Consistency of argument order made explicit using a parameter, or semantic feature for file operations

- Feature Possible values

```
Op = copy; move; link
```

- Rules

```
file-op[Op]::=   command[Op] + filename  + filename
               | command[Op] + filenames + directory

command[Op = copy] ::= cp
command[Op = move] ::= mv
command[Op = link] ::= ln
```

Other uses of TAG

- User's existing knowledge
- Congruence between features and commands
- These are modeled as derived rules

Physical models

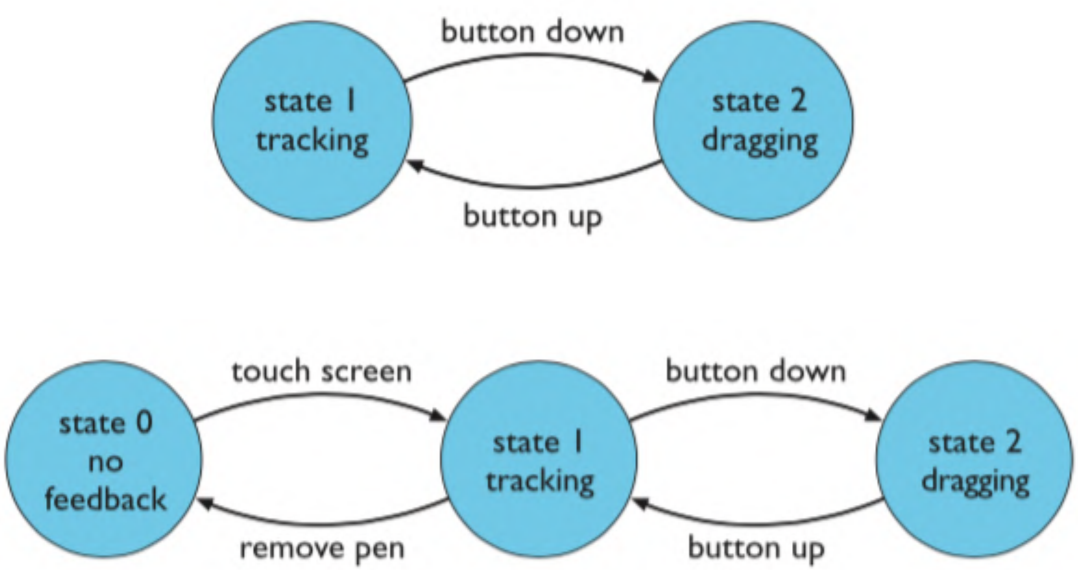
Physical and device models

- The **K**eystroke **L**evel **M**odel (**KLM**)
- Buxton's 3-state model

- Based on empirical knowledge of human motor system
- User's task: acquisition then execution.
  - these only address **execution**
- Complementary with goal hierarchies

Physical and device models

- Buxton's 3-state model



Physical and device models

- Buxton's 3-state model and Fitts' law: $a + b \log_2(D/S + 1)$

Table 12.2 Fitts' law coefficients (after MacKenzie, Sellen and Buxton [221], © 1991 ACM, Inc. Reprinted by permission)

	Device	*a* (ms)	*b* (ms/bit)
Pointing (state 1)			
	Mouse	–107	223
	Trackball	75	300
Dragging (state 2)			
	Mouse	135	249
	Trackball	–349	688

Mouse

$$\text{P[to menu bar]} = -107 + 223 \log_2(11) = 664 \text{ ms}$$
$$\text{P[to option]} = 135 + 249 \log_2(5) = 713 \text{ ms}$$

Trackball

$$\text{P[to menu bar]} = 75 + 300 \log_2(11) = 1113 \text{ ms}$$
$$\text{P[to option]} = -349 + 688 \log_2(5) = 1248 \text{ ms}$$

Keystroke Level Model (KLM)

- Lowest level of (original) **GOMS**
- Six execution phase operators
  - **Physical motor**: **K** - keystroking
    P - pointing
    H - homing
    D - drawing
  - **Mental** **M** - mental preparation
  - **System** **R** - response
- Times (**T**) are empirically determined.

$$T_{execute} = T_K + T_P + T_H + T_D + T_M + T_R$$

KLM example

```
GOAL: ICONISE-WINDOW
[select
   GOAL: USE-CLOSE-METHOD
   .     MOVE-MOUSE-TO- FILE-MENU
   .     PULL-DOWN-FILE-MENU
   .     CLICK-OVER-CLOSE-OPTION
   GOAL: USE-CTRL-W-METHOD
         PRESS-CONTROL-W-KEY]
```

- **Compare alternatives:**
  - USE-CTRL-W-METHOD vs.
  - USE-CLOSE-METHOD
- **Assume hand starts on mouse**

USE-CTRL-W-METHOD		USE-CLOSE-METHOD	
H[to kbd]	0.40	P[to menu]	1.1
M	1.35	B[LEFT down]	0.1
K[ctrlW key]	0.28	M	1.35
		P[to option]	1.1
		B[LEFT up]	0.1
Total	**2.03 s**	**Total**	**3.75 s**

Table 12.1 Times for various operators in the keystroke-level model (adapted from Card, Moran and Newell [56], published and reprinted by permission of Lawrence Erlbaum Associates, Inc.)

Operator	Remarks	Time (s)
K	Press key	
	good typist (90 wpm)	0.12
	poor typist (40 wpm)	0.28
	non-typist	1.20
B	Mouse button press	
	down or up	0.10
	click	0.20
P	Point with mouse	
	Fitts' law	$0.1 \log_2(D/S + 0.5)$
	average movement	1.10
H	Home hands to and from keyboard	0.40
D	Drawing – domain dependent	–
M	Mentally prepare	1.35
R	Response from system – measure	–

wpm = words per minute

Heurísticas do Modelo KLM

- **Identificar as ações e colocá-las como uma sequência de letras K, P, B ou H**
- **Heurísticas para Colocação de Operadores Mentais (M)**
  - **Regra 0 — Inserção Inicial de Operadores Candidatos *M***
    - Inserir ***M*** antes de todos os ***K*** *ou* ***B*** *que representam entradas do utilizador*.
    - Inserir ***M*** antes de todo ***P*** que representa um comando ou inicia uma sequência de manipulação direta.
  - **Regra 1 — Remoção de *Ms* Antecipados**
    - Se um **M** está entre dois operadores que <u>variam muito de duração</u>, então este **M** deve ser eliminado. É assumido que enquanto realiza a primeira operação ele tem tempo de pensar na segunda operação
    - **Exemplo**: **PMK** torna-se **PK**, e **PMBB** torna-se **PBB** (o clique é antecipado enquanto o mouse está sendo movido)
  - **Regra 2 — Remoção de *Ms* dentro de unidades cognitivas**
    - Se uma sequência de **K** forma uma unidade cognitiva (nome de um comando ou argumento), então remover todos os Ms exceto o primeiro.
    - **Exemplo**: Se o comando dir é representado por **MKMKMK**, a sequência correta torna-se **MKKK**

Heurísticas do Modelo KLM

- **Heurísticas para Colocação de Operadores Mentais (continuação)**
  - **Regra 3 — Remoção de *Ms* anteriores a delimitadores consecutivos**
    - Se ***K*** é um delimitador redundante no fim de uma unidade cognitiva (comando), por exemplo um delimitador de um comando imediatamente seguido do delimitador do seu argumento, então remover o ***M***.
  - **Regra 4 — Remoção de *Ms* que são delimitadores de comandos**
    - Se ***K*** é um delimitador de um comando então apagar o *M* em frente
    - Senão:
    - Se o ***K*** é um delimitador para um argumento (valor fornecido pelo usuário) ou alguma sequência que pode variar manter o *M* em frente
  - **Regra 5 — Remoção de *Ms* sobrepostos**
    - Não contar os ***M*** após ***R***
    - **Exemplo**: um tempo de espera em que o usuário aguarda uma resposta do sistema

Exemplo de Aplicação do KLM

- Mover a mão para o mouse
  H
- Apontar para o botão apropriado
  HP
- Clicar no botão de rádio
  HPBB
- Apontar para a edit box
  HPBBP
- Clicar na edit box
  HPBBPBB
- Mover a mão para o teclado
  HPBBPBBH
- Digitar a temperatura (“37.8”)
  HPBBPBBHKKKK
- Digitar Enter
  HPBBPBBHKKKKK

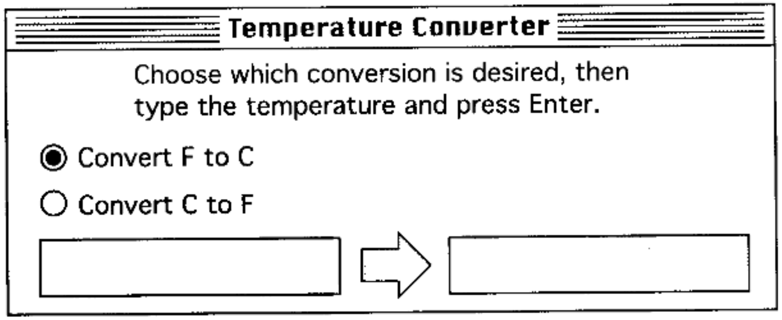
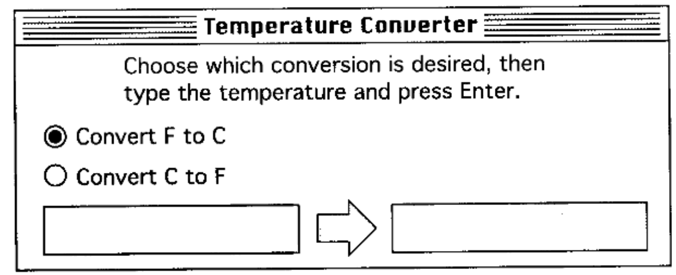


Table 12.1 Times for various operators in the keystroke-level model (adapted from Card, Moran and Newell [56], published and reprinted by permission of Lawrence Erlbaum Associates, Inc.)

Operator	Remarks	Time (s)
K	Press key	
	good typist (90 wpm)	0.12
	poor typist (40 wpm)	0.28
	non-typist	1.20
B	Mouse button press	
	down or up	0.10
	click	0.20
P	Point with mouse	
	Fitts' law	$0.1 \log_2(D/S + 0.5)$
	average movement	1.10
H	Home hands to and from keyboard	0.40
D	Drawing – domain dependent	–
M	Mentally prepare	1.35
R	Response from system – measure	–

wpm = words per minute

Exemplo de Aplicação do KLM



- **Aplicação das Heurísticas**
  - Aplicando a Regra 0:

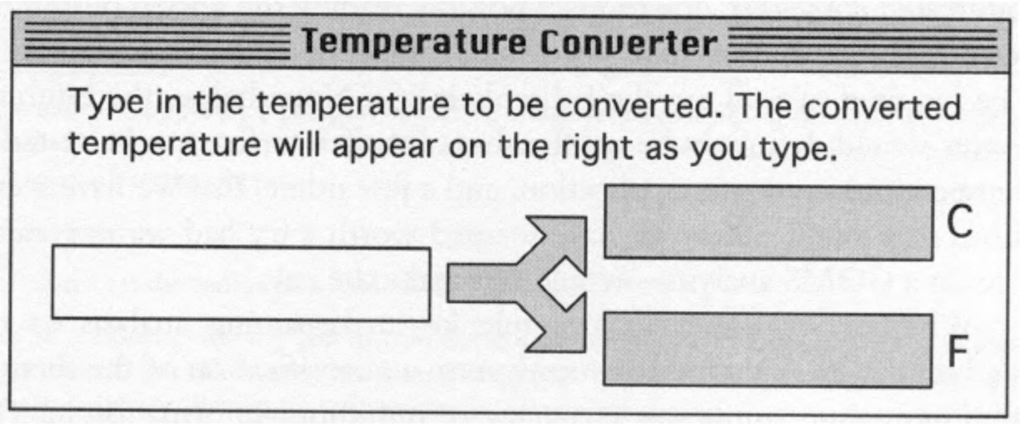
    HMPMBBPBBHMKMKMKMKMK
  - Aplicando a Regra 1: (PMK=PK, PMB = PB)

    HMPBBPBBHMKMKMKMKMK
  - Aplicando a Regra 2:

    HMPBBPBBHMKKKKMK
  - O **M** antes do último **K** tem que ser mantido pela regra 4 e as regras 3 e 5 não se aplicam neste exemplo
  - Substituindo os operadores pelos valores esperados

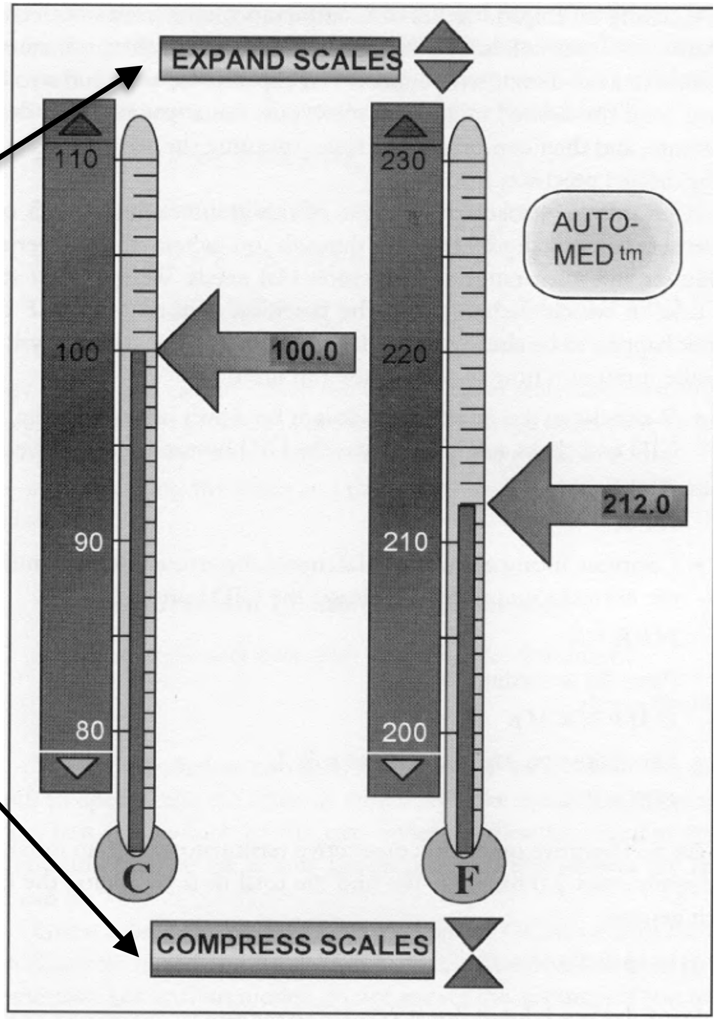
    0.4 + 1.35 + 1.1 + 0.2 + 1.1 + 0.2 + 0.4 + 1.35 + 4*(0.2) + 1.35 + 0.2 = 8.45s
  - No caso em que a conversão já está corretamente selecionada o método é:
    - MKKKKMK = 3.7s
  - Como ambas as conversões são equiprováveis:
    - (8.45+3.7)/2=6.075s

Outras Alternativas



MPKKKK 1.35 + 1.1 + 4*(0.2) = 3.25s

Outras Alternativas



Controlos

MPB 1.35 + 1.1 + 0.1 = 2.46s

or

MHPB 1.35 + 0.4 + 1.1 + 0.1 = 2.86s