Einführung in die Kognitive Modellierung mit ACT-R

Sven Brüssow Daniel Holt

Psychologisches Institut
Universität Heidelberg

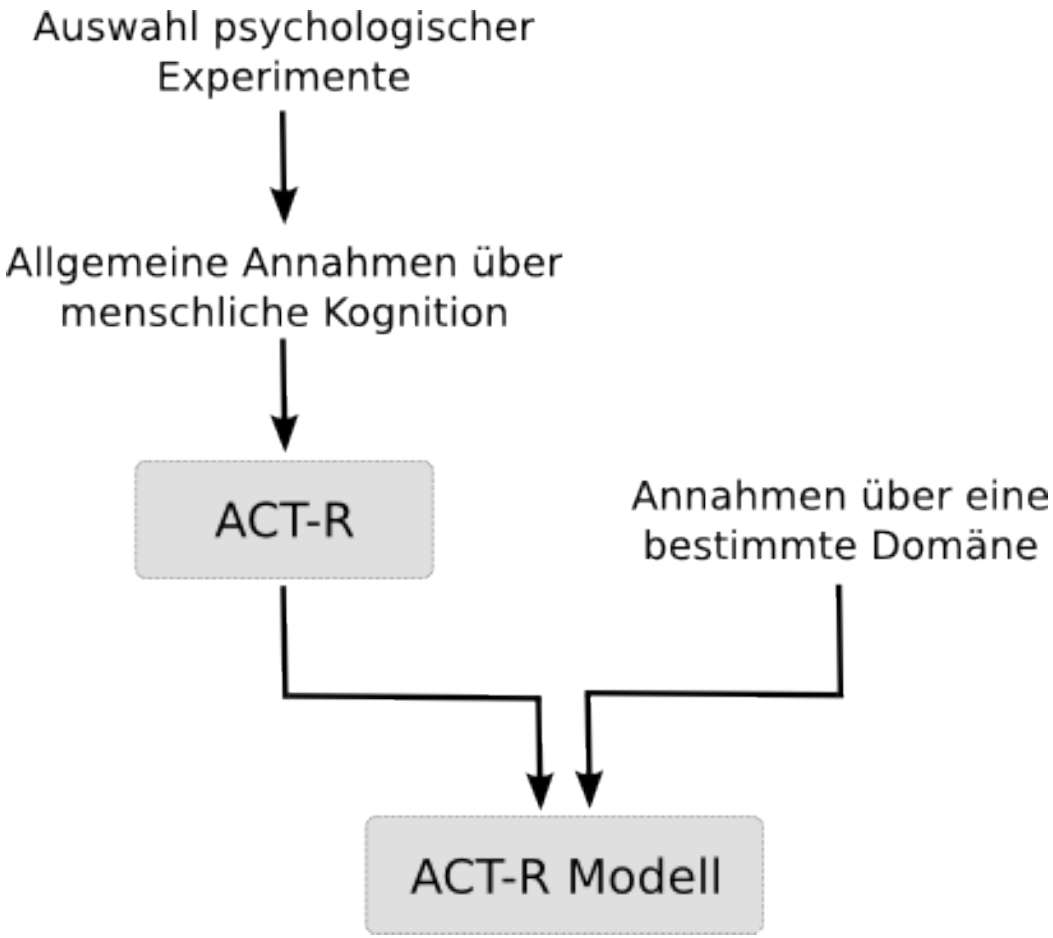
Wintersemester 2007/2008

24. Oktober 2007

Wiederholung

- Allen Newell versteht Produktionssysteme als detailierte Modelle menschlicher Kontrollstruktur.
- ACT-R ist eine kognitive Architektur auf der Basis eines Produktionssystems.
- Ein kognitivesModell ähnelt einem Programm geschrieben in der Sprache der kognitiven Architektur.

Wiederholung



vgl. http://act-r.psy.cmu.edu/about/

Die Struktur von Wissen

Zwei Arten von Wissen

- Eine besondere Rolle im Zusammenhang mit Produktionssystemen spielt die Struktur von Wissen.

- ACT-R ist eine Theorie die versucht menschliche Kognition über bestimmete Wissensstrukturen zu erklären.

ACT-R kennt zwei Arten von Wissen:

1. Deklaratives Wissen
2. Prozedurales Wissen

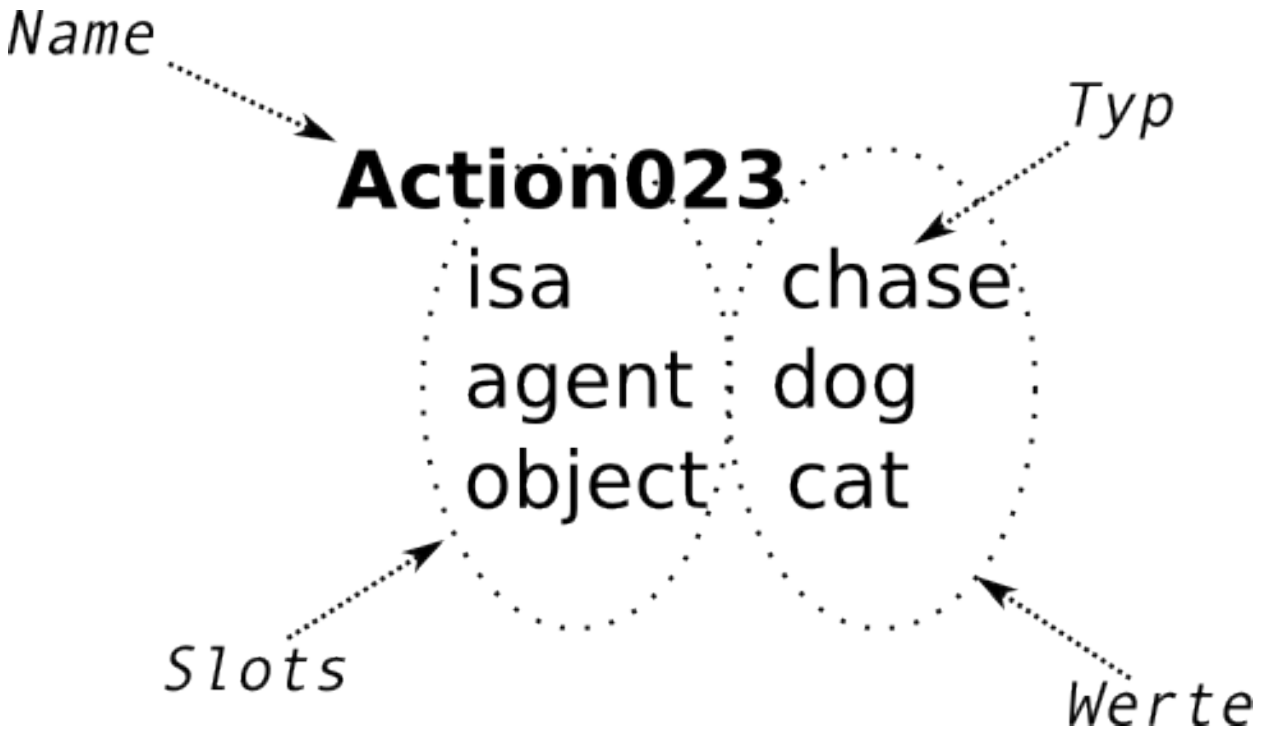
Analog kann man sagen:

1. ACT-R besitzt ein deklaratives Gedächtnis für Fakten.
2. ACT-R besitzt ein prozedurales Gedächtnis für Regeln.

Deklaratives Wissen – Chunks

- Deklaratives Wissen wird in ACT-R durch Chunks repräsentiert.
- Ein Chunk ist durch seinen Typ und seine Slots definiert.
- Der Typ steht für die Kategorie zu der der Chunk gehört.
- Slots entsprechen den Attributen einer Kategorie.
- Jeder Chunk hat einen eindeutigen Namen mit denen er referenziert werden kann.

Deklaratives Wissen – Chunks (2)



Deklaratives Wissen – Chunks (3)

Action023

ISA	chase
agent	dog
object	cat

Frau7

ISA	human
height	170
age	30
sex	female

task-2b-2t1

ISA	task
location	cafe
arrival	1100
duration	60

Fact3+4

isa	addition-fact
addend1	three
addend2	four
sum	seven

Deklaratives Wissen – Chunks (4)

```
1  (chunk-type animal class)
2
3  (add-dm                          ;; add to declarative memory
4    (mammalia isa chunk)
5    (insecta  isa chunk)
6
7    (animal-1 isa   animal
8              class mammalia)
9
10   (animal-2 isa   animal
11             class insecta)
12 )
```

Prozedurales Wissen – Produktionen (*revisited*)

- Prozedurales Wissen spiegelt sich in Verhalten wider und wird durch Produktionen repräsentiert.
- Produktionen haben einen Bedingungsteil (LHS für *left-hand side*) und einen Aktionsteil (RHS für *right-hand side*).

IF **the goal is to classify a person**
and he is unmarried
THEN **classify him as a bachelor**

Prozedurales Wissen – Produktionen (*revisited*)

- Produktionen sind Bedingungs-Aktions-Paare.
- Wenn die Bedingung (LHS) erfüllt ist, werden die Aktionen des Aktionsteils (RHS) ausgeführt, d.h. die Produktion feuert.
- Prüfen auf der LHS und Ausführen auf der RHS erfolgt zyklisch (*recognize-act cycle*).
- Es kann immer nur eine Produktion feuern!

Module und Buffer

Module und Buffer

Die LHS einer Produktion

- testet den Inhalt von Buffern
- überprüft den Zustand von Buffern und Modulen

Die RHS einer Produktion

- modifiziert den Inhalt der Buffer.
- stellt Anfragen an Module.

Jeder Buffer kann genau einen Chunk aufnehmen!

Module und Buffer (2)

Ein Modul nimmt Bezug auf eine kognitive Fähigkeit und kann typischerweise einer bestimmten Gehirnregion bzw. bestimmten Gehirnregionen zugeschrieben werden.

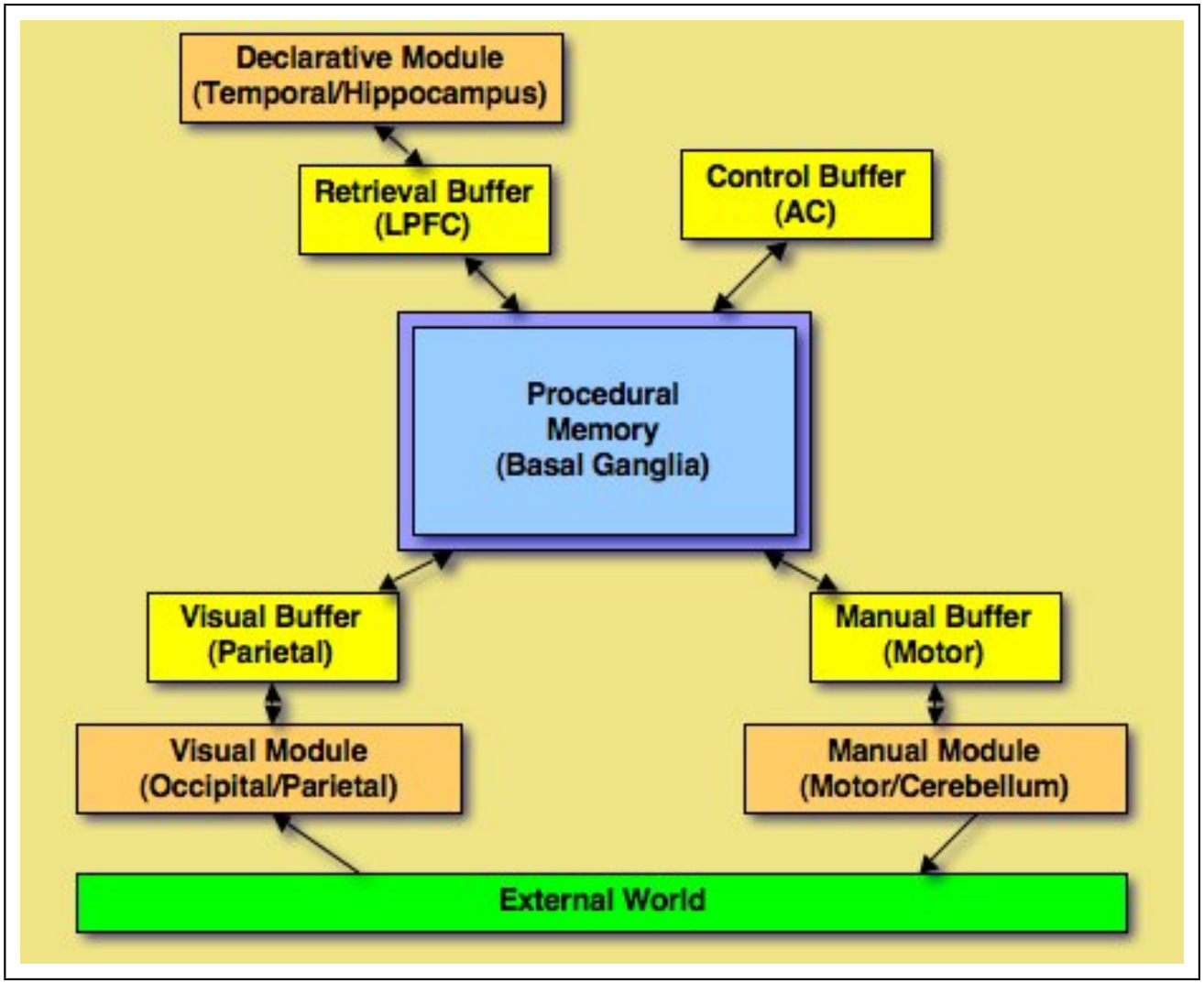
In diesem Zusammenhang gibt 2 Arten von Modulen:*

1. Module der Wahrnehmung und Motorik
2. Module der internen Repräsentation

Ein Buffer ist eine Schnittstelle zwischem dem prozeduralen Modul und einem anderen Modul.

*ACT-R kennt weitere Module, die aber nicht Teil der Theorie sind.

Exkurs: Module, Buffer und neuronale Korrelate



http://tcw2.ppsw.rug.nl/~niels/overview.html

```
#####################################
ACT-R Version Information:
Framework               : 1.2 [r505]
CENTRAL-PARAMETERS      : 1.0       a module that maintains parameters used by other modules
IMAGINAL                : 1.1       The imaginal module provides a goal style buffer with a delay and
                                    an action buffer for manipulating the imaginal chunk
PRINTING-MODULE         : 1.0       Coordinates output of the model.
BUFFER-TRACE            : 1.0       A module that provides a buffer based tracing mechanism.
NAMING-MODULE           : 1.2       Provides safe and repeatable new name generation for models.
BOLD                    : 1.1       A module to produce BOLD response pbrownictions from buffer request
                                    activity.
MOTOR                   : 2.3       Module to provide a model with virtual hands
SPEECH                  : 2.2       A module to provide a model with the ability to speak
VISION                  : 2.4       A module to provide a model with a visual attention system
AUDIO                   : 2.3       A module which gives the model an auditory attentional system
RANDOM-MODULE           : 1.0       Provide a good and consistent source of pseudorandom numbers for all
                                    systems
BUFFER-PARAMS           : 1.0       Module to hold and control the buffer parameters
ENVIRONMENT             : 2.0       A module to handle the environment connection if opened
PRODUCTION-COMPILATION: 1.1         A module that assists the primary procedural module with compiling
                                    productions
DEVICE                  : 1.1       The device interface for a model
PROCEDURAL              : 1.3       The procedural module handles production definition and execution
DECLARATIVE             : 1.1       The declarative memory module stores chunks from the buffers for
                                    retrieval
GOAL                    : 1.1       The goal module creates new goals for the goal buffer
UTILITY                 : 2.0       A module that computes production utilities

######### Loading of ACT-R 6 is complete #########
Welcome to OpenMCL Version 1.1-pre-061231 (DarwinX8664)!
?
```

Module und Buffer (3)

- Ein Modul führt Aktionen aus, die über dessen Buffer angefordert werden.
- Über einen Buffer werden Anfragen über den Status des betreffenden Moduls gestellt.
- Ein Buffer kann die Kopie eines Chunks aufnehmen, die dann für alle anderen Module sichtbar ist. Das Original bleibt unberührt!
- Jedes Modul hat Zugriff auf jeden der Chunks in den Buffern. Typischerweise greift ein Modul nur auf den Chunk in seinem eigenen Buffer zu.

Module und Buffer (4)

- ACT-R stellt das prozedurales und deklaratives Wissen über das prozedurale und deklarative Modul zur verfügung.

- Das deklarative Modul kommuniziert mit dem prozeduralen Modul über den *retrieval buffer*.

Module und Buffer (5)

Neben dem deklarativen und prozeduralem Modul gibt es weitere Module, die wie das deklarative Modul mit dem prozeduralen Modul über ihre Buffer kommunizieren:

Modul	**Art**	**Buffer**
procedural	interne Repräsentation	(kein eigener Buffer)
declarative	interne Repräsentation	retrieval
intentional	interne Repräsentation	goal
imaginal	interne Repräsentation	imaginal
visual	Wahrnehmung	visual
auditory	Wahrnehmung	aural
manual	Motorik	manual

Erste Schritte

Notation

```
(p Name "optional documentation string"
   buffer tests
==>
   buffer changes and requests
)
```

- Tests bestehen aus einer Anzahl von Mustern auf der LHS vor dem ==>, die mit dem Inhalt des Buffers verglichen werden.
- Bei Übereinstimmung werden die Aktionen auf der RHS nach dem ==> ausgeführt bzw. die Produktion „feuert“.

Notation – LHS: Tests und Statusabfragen

Mit `=buffername>` kann der Inhalt überprüft werden:

```
=goal>                      ;; Wenn das Ziel
    isa     selection       ;; vom Typ 'selection' ist und
    task    request         ;; der slot 'task' den Wert 'request' hat
```

Mit `?buffername>` kann der Status abgefragt werden:

```
?retrieval>         ;; Wenn ueber den deklarativen Buffer
   state   error    ;; erfolglos versucht wurde einen Chunk abzurufen
```

Der `state`-Slot kann die Werte `full`, `empty`, `requested`, `urequested`, `free`, `busy` und `error` enthalten.

Notation – RHS: Anfragen und Änderungen

Anfragen bestehen aus einer Anzahl von Mustern auf der RHS nach dem ==> und werden mit +buffername> eingeleitet.

```
+retrieval>              ;; dann frage das deklarative Modul
    isa      animal      ;; nach einem Chunk vom Typ 'animal'
```

Änderungen bestehen aus einer Anzahl von Mustern auf der RHS und werden mit =buffername> eingeleitet.

```
=goal>                   ;; dann aendere vom aktuellen Ziel
    task     harvest     ;; im Slot 'task' den Wert zu 'harvest'
```

Notation

```
(p request-animal
  =goal>
    isa           selection
    task          request
==>
  =goal>
    task          harvest
  +retrieval>
      isa         animal
)
```

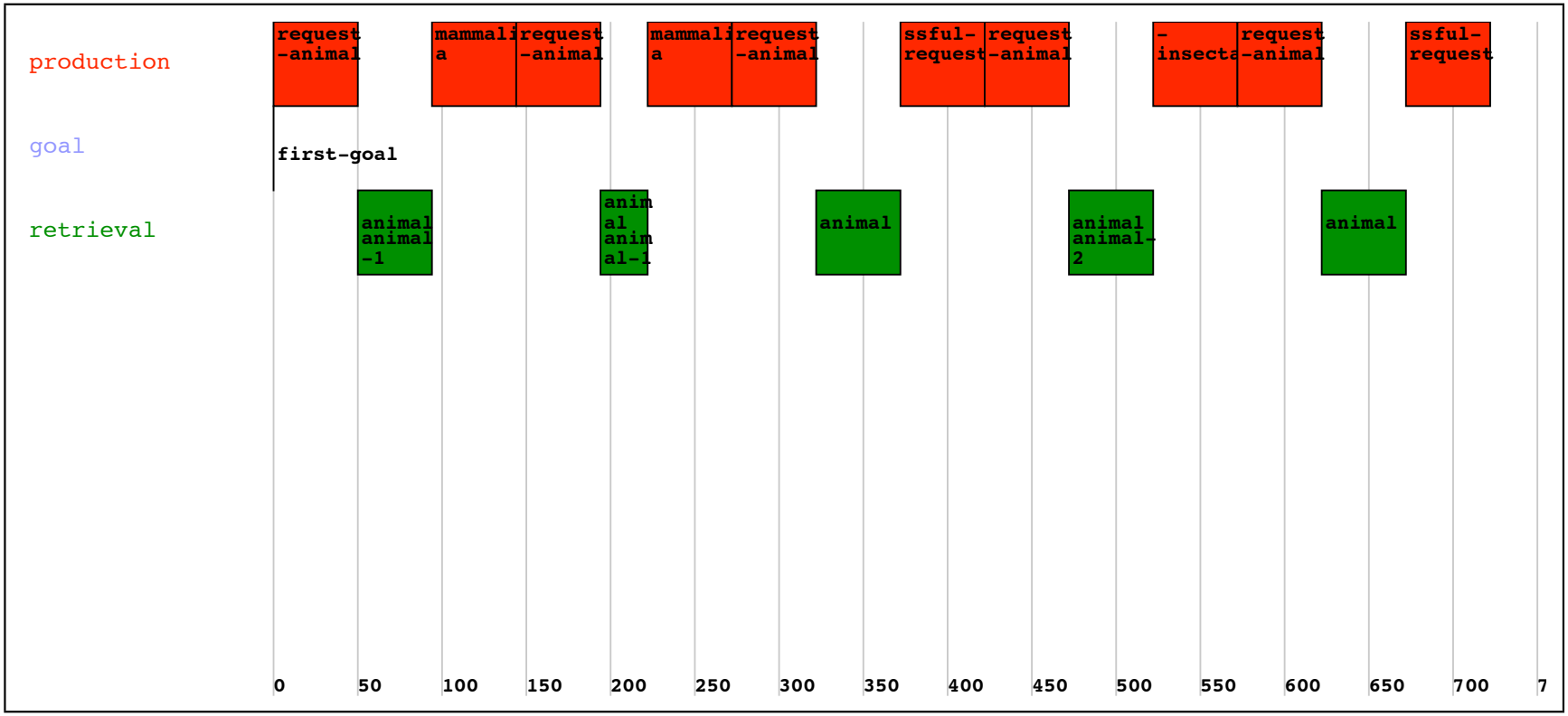
```
(p harvest-insecta
  =goal>
    isa             selection
    task            harvest
  =retrieval>
    isa           animal
    class           insecta
==>
  =goal>
    task          request

  !output! "insecta"
)
```

Notation

```
1  (p unsuccessful-request
2    =goal>
3      isa         selection
4      task         harvest
5    ?retrieval>       ;; query state of buffer
6      state        error
7  ==>
8    =goal>
9      task         request
10
11   !output! "*** Failed! ***"
12 )
```

```
? (load "animal-request.lisp")
#P"/Users/bruessow/Documents/BMBF-PS/ACT-R-Seminar/models/animal-request.lisp"
? (run .75)
     0.000   GOAL                 SET-BUFFER-CHUNK GOAL FIRST-GOAL REQUESTED NIL
     0.050   PROCEDURAL           PRODUCTION-FIRED REQUEST-ANIMAL
     0.094   DECLARATIVE          SET-BUFFER-CHUNK RETRIEVAL ANIMAL-1
     0.144   PROCEDURAL           PRODUCTION-FIRED HARVEST-MAMMALIA
"mammal"
     0.194   PROCEDURAL           PRODUCTION-FIRED REQUEST-ANIMAL
     0.222   DECLARATIVE          SET-BUFFER-CHUNK RETRIEVAL ANIMAL-1
     0.272   PROCEDURAL           PRODUCTION-FIRED HARVEST-MAMMALIA
"mammal"
     0.322   PROCEDURAL           PRODUCTION-FIRED REQUEST-ANIMAL
     0.372   DECLARATIVE          RETRIEVAL-FAILURE
     0.422   PROCEDURAL           PRODUCTION-FIRED UNSUCCESSFUL-REQUEST
"*** Unsuccessful request! ***"
     0.472   PROCEDURAL           PRODUCTION-FIRED REQUEST-ANIMAL
     0.522   DECLARATIVE          SET-BUFFER-CHUNK RETRIEVAL ANIMAL-2
     0.572   PROCEDURAL           PRODUCTION-FIRED HARVEST-INSECTA
"insecta"
     0.622   PROCEDURAL           PRODUCTION-FIRED REQUEST-ANIMAL
     0.672   DECLARATIVE          RETRIEVAL-FAILURE
     0.722   PROCEDURAL           PRODUCTION-FIRED UNSUCCESSFUL-REQUEST
"*** Unsuccessful request! ***"
     0.750   ------               Stopped because time limit reached
0.75
102
NIL
?
```

Buffer Trace



Notation – Variablen

- Das = Präfix kennzeichnet eine Variable.
- Variablen testen auf generelle Bedingungen.
- Variablen haben zwei grundlegende Einsatzmöglichkeiten:
  1. In der Bedingung könne sie zwei oder mehr Slots miteinander vergleichen, ohne den genauen Wert zu kennen.
  2. Sie können Werte von einem Slot zu einem anderen Slot kopieren.

Notation

```
1   (p harvest-animal
2     =goal>
3     isa             selection
4     task            harvest
5     =retrieval>
6     isa           animal
7     class         =class
8   ==>
9     =goal>
10    task          request
11    !output! ("Animal: ~S~%Class : ~S" =retrieval =class)
12  )
```

Aufgaben für nächste Woche:

a. ACT-R installieren
b. Unit 1 des ACT-R Tutoriums lesen
c. count-Modell