

Referential nets as ACT-R declarative memory representation

Markus Guhe

m.guhe@ed.ac.uk

Human Communication Research Centre & Linguistics and English Language, University of Edinburgh



Problem

Unified theories of cognition neglect language.

Goal

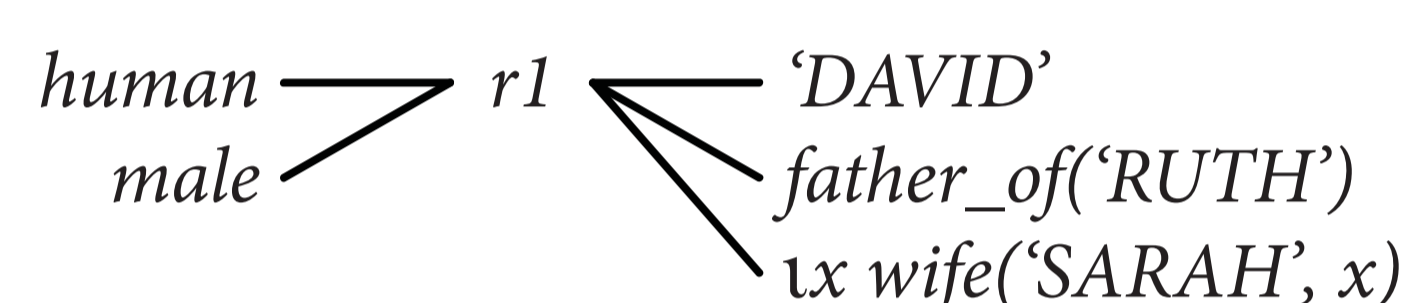
- To bridge the gap between computational linguistics and unified theories of cognition.
- To be able to use results from computational linguistics in unified theories of cognition.

Method

Take a representation for representing multiple levels of language (with an emphasis on discourse) and translate it into ACT-R chunks.

Referential Nets

- Referential nets (Habel 1986) are similar to DRT (Kamp & Reyle 1993).
- They are better suited than DRT for achieving the goal – they emphasize the referential nature of language over the truth-values of utterances.
- Referential objects (refOs) are terms.
- They are specified by typed expressions: attributes or designations.
- Example:



- Two attributes (*human*, *male*)
- Three designations:
 - Names ('DAVID')
 - Functional expressions (*father_of('RUTH')*)
 - Descriptions (*ix wife('SARAH', x)*; 'Sarah is the wife of the entity represented by r1')

ACT-R

- Chunks in ACT-R are lists of slot–value pairs (Anderson & Lebiere 1998: 23):

```
(fact3+4 ISA addition-fact
 addend1 three
 addend2 four
 sum seven)
```

- The corresponding chunk-type:
(chunk-type addition-fact addend1 addend2 sum)

Basic translation

- Chunks in ACT-R and expressions in referential nets are typed.
- Names as chunks:
(chunk-type name-fact name)
(name-fact1 ISA name-fact name david)
- Descriptions as chunks (refNet sort of wife is <person, person>):
(chunk-type wife person1 person2)
- Abstracted descriptions:
(chunk-type abs-designation op var predicate)
op: ι , η , some-t, all-t
var: abstraction variable
predicate: reference to a chunk containing the predicate

Problem 1 & solution: list attributes

- Main problem: referential nets freely combine expressions in a principled manner.
- This is difficult to realise in ACT-R.
- Problem 1: attributes can have values, which can be lists, e.g. for David:
member_of(['GRP1', 'GRP2'])
- Solution: split the list attribute into multiple attributes:
member_of('GRP1')
member_of('GRP2')

Problem 2: assignment of expressions

- Problem 2: translating the assignment of expressions to referential objects
- Example: human is an attribute of r1.
- The problem: refOs are not restricted in number or kinds of attributes and designations.

Solution 1

- Define a general chunk-type and distribute the information over multiple chunks:
(chunk-type attrib-assign refo attrib)
(att1 ISA attrib-assign refo r1 attrib human)
(att2 ISA attrib-assign refo r1 attrib male)
- The knowledge about the entity represented by r1 cannot be accessed at once.
- Multiple chunk retrievals are needed to find the required information.

Solution 2

- Store the attributes in a list and have just one chunk per refO:
(chunk-type att-assig refo att)
(assig1 ISA att-assig refo r1 att '(human male))
- All information is available at once.

Two different predictions

- Solution 1 predicts that accessing knowledge about a refO takes time.
- Solution 2 finishes the comparison within a single ACT-R cycle.
- ACT-R predicts behaviour according to solution 1.
- Referential nets predict solution 2 (immediate access to a refO's information).
- Solution 2 is supported by *INC* (*incremental conceptualiser*, Guhe et al 2003).
- Processing of concepts predicts different processing times:
 - Solution 1 predicts longer processing times for bigger refOs: at least one ACT-R cycle per attribute.
 - No empirical support for this prediction: there is no evidence that familiar (big) concepts take longer to process.

A first implementation

In ACT-R 6 the function computing the similarity between two chunks has had to be extended to allow for lists:

```
'(human male) x '(human) = 1/2
'(human male) x '(human woman) = 1/3
```

Discussion

- There are many more problems, e.g. arguments of expressions can be refOs.
- Overall, it seems possible to translate referential nets into ACT-R chunks.
- ACT-R 6:
 - Provides a framework for implementing simultaneous access to a refO's information.
 - Supports more detailed comparisons within working models.
- Next step: partial reimplementing of *INC* in ACT-R 6 to test the predictions.
- The difficulties caused by the limitations of ACT-R's chunks equally affect DRT.
- The problems discussed here are not limited to referential nets but indicate general obstacles in transferring CL representations into ACT-R.

Acknowledgments

Thanks to Mike Schoelles for discussions of ACT-R and to Ellen Bard and Mark Steedman for comments. This research was supported by grant NSF-IIS-0416128.

Literature

- Anderson, JR, & Lebiere C (1998) *The atomic components of thought*. Erlbaum.
- Budiu, R and Anderson, JR (2004). Interpretation-based processing: a unified theory of semantic sentence comprehension. *Cognitive Science*, 28(1):1–44.
- Guhe, M (in press) *Incremental conceptualisation for language production*. Erlbaum.
- Guhe, M, Habel, C & Tschander, L (2003) Incremental production of preverbal messages with *INC*. In: *Proceedings of ICCM-5*, 123–128.
- Habel, C (1986) *Prinzipien der Referentialitat*. Springer.
- Kamp, H & Reyle, U (1993). *From discourse to logic*. Kluwer.
- Landauer, TK, Foltz, PF & Laham, D (1996). An introduction to latent semantic analysis. *Discourse Processes*, 25(2/3): 259–284.
- Lewis, RL (1993). *An architecturally-based theory of human sentence comprehension*. PhD thesis, CMU.
- Newell, A (1990). *Unified theories of cognition*. Harvard University Press.
- Reiter, E (1994). Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In: *Proceedings of INLGW-7*, 163–170.

