A relaciokrol csinaltam egy kis osszefoglalot:

properties of relations:

all relations have 1 or 2 kinds of arguments (not more).

S,D symmetric, p gets transferred. Similar, Identical

AND,OR,NOT symmetric, some p rules apply to compound And, Or, Not

C asymetric, p trasfer from class children Class

F asymetric, implied C relation: C(a,F(a,b)) Feature

Q asymetric, implied C relation: C(b,Q(a,b)), g transfer Quantifier

A asymetric, implied C relation: C(b,A(a,b)) Action

I asymetric, implied C relation: C(a,I(a,b)) Impact (on object)

P asymetric, no C relation, no p transfer Part or Possession

R asymetric, no C relation, no p transfer Relative (space etc)

T asymetric, no C relation, no p transfer Time

M ordered, no C relation, no p transfer More

IM asymetric, no C relation, yes p transfer Implication

N asymetric, no C relation, yes p transfer Necessary condition

V asymetric, no C relation, no p transfer Relevant

Egy pelda input kerdesekkel. Par dolog el van hanyagolva az egyszerusites kedveert. Peldaul a my, every nevmasok.

1. “any concept” placeholder

Ez egy pelda arra, hogy hasznos lehet olyan szimbolumokat hasznalni, amelyek helyere barmely concept behelyettesitheto. Ezek lennenek a %1 es a %2:

*If %1 consists of %2, then %2 is part of %1.* IM ( F(A(%1, consist), R(of,%2)) , P(%2,x))

1. Mentalese translation module content

There are two ways to achieve a certain mentalese statement. One way is to make a direct assignment between grammar parse output and mentalese, like for “part of”, and “means that”, see below. The other way is to have IM or other reasoning-statements, like for “consist of”.

NP1 BE PARTOF NP2 🡪 P(NP1, NP2) (something is part of something)

S1 MEAN S2 🡪 IM(S1,S2) S(S1,S2) (driving a car means that you sit in the car and make it move)

**Family**  (“my family” not yet distinguished from general “family”)

A family consists of people. F(A(family, consist), R(of, people))

Family is a group. C(family,group)

If a person is part of a group, then the person is member of the group.

IM ( P(person,group), D(person, P(member,group)))

People are persons. C(people,person)

Joe is a person. C(joe,person)

Mary is a person. C(mary,person)

Father is a person. C(father,person)

Mother is a person. C(mother,person)

Father is Joe`s parent. D(father, P(parent,Joe))

Mother is Joe`s parent. D(mother, P(parent,Joe))

Mary is part of my family. P(mary, family)

This family consists of Joe, his sister and his parents.

F(A(family,consist), R(of, joe, sister, parent))

Mary is Joe`s sister. D(mary, P(sister,joe))

Bob is Joe`s friend. D(bob, P(friend,joe))

Friend is a person. C(friend,person)

Bob is not part of my family. NOT(P(bob,family))

Jim is a friend of Joe. C(jim,P(friend,joe))

What does a group consist of? F(A(group,consist), R(of,?)) - “what” is lost

Is Joe a person? C(Joe,person)?

What is Joe? C(Joe, ?)

Is Joe part of my family? P(Joe, family)?

What is Joe part of? P(Joe, ?)

Is Joe member of my family? D(Joe, P(member, family))?

Is Mary part of my family? P(mary,family)?

Is father part of my family? P(father,family)?

Who is part of my family? P(?, family)

Who are the members of my family? D(P(member,family),?)

Is Bob part of my family? P(bob,family)?

Is Jim part of my family? P(jim,family)?

**General versus specific, my, his, etc**

1.

In addition to p and u, we introduce a third dimension, g. (generality). Every concept has a g.

2.

We change the mentalese translation:

Mary is part of my family. P(mary, Q(my,family)), g[my]=0

This g[my]=0 is that we set a concept property, g, to a number, 0.

This is a new functionality in mentalese translation.

Implementation details:

3.

Q relation (qualifier) involves some special code for reasoning. It makes use of one more truth-dimension, g, for generality.

Q-reasoning is this: if we have Q(a,b), and for concept “a” we have g value of g1 -> then we modify g value of “b” depending on g1 (in the simplest case it becomes g1). (p of Q(a,b) should also play a role.)

4.

g-value processing implementation (simplest case): let us take the “family” example:

* The word “family” has a first meaning, the “f” concept, this concept will have g=100, maximum possible generality. All word-meaning concepts are stored with maximum generality.
* Upon mentioning a word and storing to WM, by default we take g=100.
* After processing Q(my,family), we have a family-concept in WM, let it be f2. It is linked to f, but f2`s g-value comes from “my” which has g=0, so let us set g[f2]=0. (But for “a family” we will have g=100.)
* **G1 : Generality based reasoning** generates the concept C(f2,f) in WM.
* **G2 : for relation P,** if we have P(Q(a,b),c) then g[b] has to impact g[c], in simplest case making it equal. And P(Q(a,b),c) generates P(b,c) which also has a g, in simplest case g[b].
* **G3 : for some X relations,** if we have X(b,c) with high generality, and we have C(b1,b) where g[b1]=0, then based on C we reason X(b1,c), but now we transform this to X(b1,c1) and generate C(c1,c), and g[c1]=0. So we conclude that Bob has a **specific** family.
* **G4 : for relation C,** if we have C(f2,f) and C(f3,f) and g[f2]=0, g[f3]=0, then this generates that D(f2,f3) has p=0, which means that f2 and f3 are not the same (they are different instances of class family, without overlap).

5.

Question-answering needs to be made more sophisticated making use of g.

So far we had:

A boy is a person. C(boy,person), boy, person

Bob is a boy. C(bob,boy), bob, boy

Bob is a person? C(bob,person)? Answer: C(bob,person) p=100.

Now this becomes more complicated to make a difference between general concepts and their specific instances:

A boy is a person. C(boy,person) p=100. boy g=100, person g=100

Bob is a boy. C(bob,boy) p=100. bob g=0, boy2 g=0, C(boy2,boy)

Bob is a person? C(bob,person)?

Generate: C(bob,person2), p=100, person2 g=0, C(person2,person)

Answer: yes

Now we can reason about different families:

A family consists of people. F(A(family, consist), R(of, people)) g[f1]=100

Mary is part of my family. P(mary, Q(my,family)), g[f2]=0, generated: C(f2,f)

Every person is part of a family. P(Q(every,person), family) leads to f3, g[f3]=100, g[person]=100

We reason P(person,family) with high generality

Bob is a person. C(bob,person) g[bob]=0

Is Bob part of a family? P(bob,family)?

Reasoning: 1. P(bob,family) leads to P(bob,f4) and C(f4,family)

So the answer is yes, the generated concepts are

P(bob,f4) g=0

C(f4,family) p=100, g[f4]=0, g[family]=100

D(f2,f4) p=0, Bob`s family is not Mary`s family.

**Going to places, being at palces, doing sg**

Joe goes to school every morning. F(A(joe,go), R(to,school), T(morning))

Mary goes to preschool every morning. F(A(mary,go), R(to,preschool), T(morning))

Father goes to drive the taxi early. F(A(father,go), R(to,I(drive,taxi)), T(early))

The school is a place. C(school,place)

The preschool is a place. C(preschool,place)

Drive is not a place. NOT(C(drive,place))

Father likes to drive the taxi. F(A(father,like), R(to, I(drive,taxi))

Is the school a place? C(school, place)?

What is the school? C(school, ?)

Can you mention a place? C(?, place)

What does Joe do every morning? F(A(Joe,?), T(morning))

Where does Joe go? F(A(joe,go), R(to,?))

Where does Joe go every morning? F(A(joe,go), R(to,?), T(morning))

When does Joe go to school? F(A(joe,go), R(to,school), T(?))

When does Joe go to preschool? F(A(joe,go), R(to,preschool), T(?))

Where does father go? F(A(father,go), R(to,?))

Does father drive? A(father,drive)?

What does father drive? A(father, I(drive,?))

Who drives the taxi? A(?, I(drive, taxi))