Cognitive Modeling

Analogical Reasoning, Problemsolving, and Learning

Ute Schmid

Kognitive Systeme, Angewandte Informatik, Universität Bamberg

last change: 8. Dezember 2014

Ubiquitousness of Analogical Thinking

Analogy is a powerful and often used cognitive skill:

Exploiting experience from one domain („base“)
to explain/predict unknown aspects of or solve problems
in a different domain („target“).

> *Analogy pervades all our thinking, our everyday speech and our trivial conclusions as well as artistic ways of expression and the highest scientific achievements.*
>
> *Polya, How to Solve It, 1945*

Analogy as a Central Research Topic in Cognitive Science

e.g., Gentner, D., Holyoak, K.J., and Kokinov, B. (2001). *The Analogical Mind – Perspectives from Cognitive Science*. MIT Press.

- Cognitive Simulation Models (Psychology, AI)
- Emotional Analogies, Scientific Discovery (Philosophy, Psychology, AI)
- Metaphors and Analogy (Linguistics, Philosophy, Mathematics)
- Neurocognitive Basis of Relational Reasoning (Neuropsychology)
- Analogy in Education (Psychology)
- Analogy in (Software) Engineering (HCI, AI)
- ...

Example 1: Analogy in Literature

Analogy or proportion is when the second term is to the first as the fourth to the third. We may then use the fourth for the second, or the second for the fourth. Sometimes too we qualify the metaphor by adding the term to which the proper word is relative. [...] As old age is to life, so is evening to day. Evening may therefore be called, 'the old age of the day,' and old age, 'the evening of life,' or, in the phrase of Empedocles, 'life's setting sun.'

Aristotle, Poetics, chap. 21, "Words"

Example 2: Analogy in History and Politics

End of war in Iraq – End of World War II

Global investor – locust

Die Methode, nach der Analogie zu schließen, ist, wie überall, so auch in der Geschichte ein mächtiges Hülfsmittel; aber sie muß durch einen erheblichen Zweck gerechtfertigt, und mit ebensoviel Vorsicht als Beurteilung in Ausübung gebracht werden.

Was heisst und zu welchem Ende studiert man Universalgeschichte?
Akademische Antrittsrede von Friedrich Schiller
am 26.5.1789 in Jena

Used to transport negative/positive emotions!

Example 2: Analogy in History and Politics

End of war in Iraq – End of World War II

Global investor – locust

Die Methode, nach der Analogie zu schließen, ist, wie überall, so auch in der Geschichte ein mächtiges Hülfsmittel; aber sie muß durch einen erheblichen Zweck gerechtfertigt, und mit ebensoviel Vorsicht als Beurteilung in Ausübung gebracht werden.

Was heisst und zu welchem Ende studiert man Universalgeschichte?
Akademische Antrittsrede von Friedrich Schiller
am 26.5.1789 in Jena

Used to transport negative/positive emotions!

Example 2: Analogy in History and Politics

End of war in Iraq – End of World War II

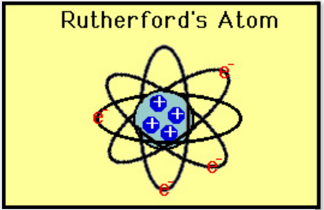
Global investor – locust

> *Die Methode, nach der Analogie zu schließen, ist, wie überall, so auch in der Geschichte ein mächtiges Hülfsmittel; aber sie muß durch einen erheblichen Zweck gerechtfertigt, und mit ebensoviel Vorsicht als Beurteilung in Ausübung gebracht werden.*
>
> *Was heisst und zu welchem Ende studiert man Universalgeschichte?*
> *Akademische Antrittsrede von Friedrich Schiller*
> *am 26.5.1789 in Jena*

Used to transport negative/positive emotions!

Example 3: Scientific Analogy

The structure of the hydrogen atom
is like
the structure of the solar system.
Rutherford, 1911



Analogy as source of scientific and artistic creativity.

Analogy in Education

Prove: the product of two even numbers is an even number

The product of two odd numbers is an odd number.
Proof:

- Odd number: *even* $+1 = 2N + 1$
- Product:
  $(2N + 1) \cdot (2M + 1) = 4NM + 2N + 2M + 1 = 2(2NM + N + M) + 1$

Analogy

- Even number: $2N$
- Product: $2N \cdot 2M = 4NM = 2(2NM)$

Analogy in Education

Prove: the product of two even numbers is an even number

The product of two odd numbers is an odd number.
Proof:

- Odd number: *even* $+1 = 2N + 1$
- Product:
  $(2N + 1) \cdot (2M + 1) = 4NM + 2N + 2M + 1 = 2(2NM + N + M) + 1$

Analogy

- Even number: $2N$
- Product: $2N \cdot 2M = 4NM = 2(2NM)$

Analogy in Education

Prove: the product of two even numbers is an even number

The product of two odd numbers is an odd number.
Proof:

- Odd number: *even* $+1 = 2N + 1$
- Product:
  $(2N + 1) \cdot (2M + 1) = 4NM + 2N + 2M + 1 = 2(2NM + N + M) + 1$

Analogy

- Even number: $2N$
- Product: $2N \cdot 2M = 4NM = 2(2NM)$

Analogy in Software Engineering

Code reuse vs developing from scratch

Systematic support of reuse can reduce development costs dramatically

Defense Information Systems Agency, 1995

In short, the business case for reuse consists of avoiding 80% of the development costs for reused components (plus some additional maintenance savings) minus the 50% extra it costs to build the reusable component in the first place.

Jeffrey Poulin and Brent Carlson, Computerworld, Feb 2004

Analogy and Similarity

	Attributes mapped to target	Relations mapped to target	Example
Mere Appearance	Many	Few	A sunflower is like the sun
Literal Similarity	Many	Many	The K5 solar system is like our solar system
Analogy	Few	Many	The atom is like our solar system
Abstraction	Few	Many	The atom is a central force system
Metaphor	x	x	*She is the sun of my live*

(Gentner, 1983, 1997)

Three Kinds of Analogy

Proportional

- „A is to B as C is to ?D"
- *Evening is to Day as Old Age is to Life* $\hookrightarrow$ *"last part of"*
- most simple form of analogy: transfer of *one* relation

Predictive/Explanatory

- Carry-over of known principles to a new domain of interest
- *Rutherford Analogy*

in Problem Solving

- *within domain*, use of examples (cf. CBR)
- Transfer of a known solution to a new problem
- Mathematical/programming problems

Three Kinds of Analogy

Proportional

- „A is to B as C is to ?D“
- *Evening is to Day as Old Age is to Life* $\hookrightarrow$ *“last part of”*
- most simple form of analogy: transfer of *one* relation

Predictive/Explanatory

- Carry-over of known principles to a new domain of interest
- *Rutherford Analogy*

in Problem Solving

- *within domain*, use of examples (cf. CBR)
- Transfer of a known solution to a new problem
- Mathematical/programming problems

Three Kinds of Analogy

Proportional

- „A is to B as C is to ?D“
- *Evening is to Day as Old Age is to Life* $\hookrightarrow$ *“last part of”*
- most simple form of analogy: transfer of *one* relation

Predictive/Explanatory

- Carry-over of known principles to a new domain of interest
- *Rutherford Analogy*

in Problem Solving

- *within domain*, use of examples (cf. CBR)
- Transfer of a known solution to a new problem
- Mathematical/programming problems

Subprocesses in Analogical Reasoning

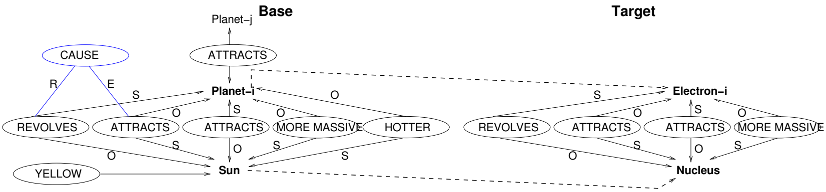
Overview

- Representation
- Retrieval
- Mapping
- Inference/Transfer
- Learning

(in all cognitive theories: Gentner, Holyoak, Keane, ...)

Representation

- Structural representation (graph, term, semantic net, ...)
- Problem: Representation crucial for mapping success
- *on(a,b)* vs. *below(b,a)*; $x > y$ vs. $x - y > 0$

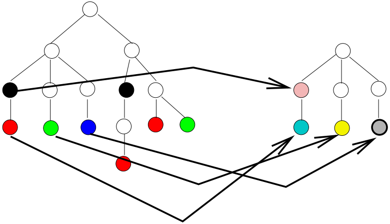


Retrieval

- Bottleneck of analogy
- Governed by superficial similarity (feature based measures)
- Novices fail to identify useful base problem (experiments by Novick, 1988)
- In Education: Present suitable base problems explicitely
- In Computation: Use structural similarity for retrieval

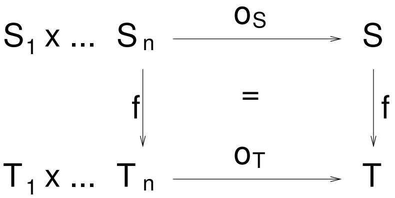
Mapping

- Core of analogy
- Structure preserving mapping
- First order (same relations/functions in both domains) or higher order
- In well-structured, formal domains: homomorphism



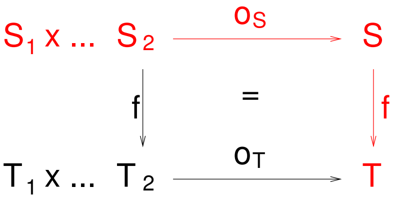
Homomorphism

- Structure preserving mapping $f : S \to T$
- such that $f(o_S(s_1, \ldots, s_n)) = o_T(f(s_1), \ldots, f(s_n))$



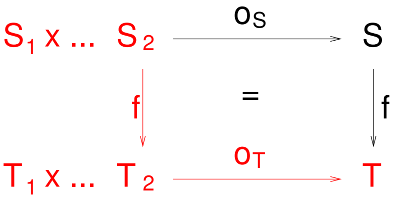
Homomorphism

- Structure preserving mapping $f : S \to T$
- such that $f(o_S(s_1, \ldots, s_n)) = o_T(f(s_1), \ldots, f(s_n))$

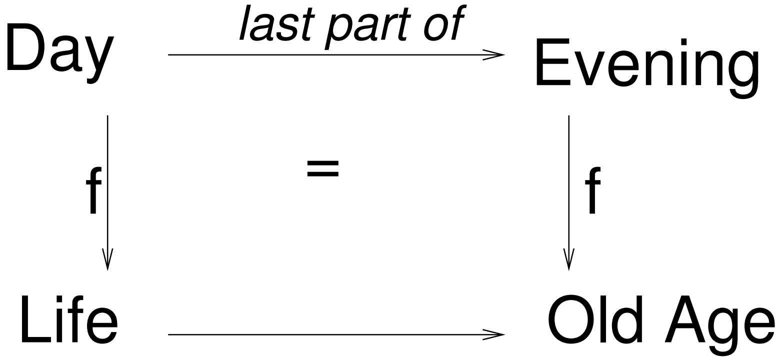


Homomorphism

- Structure preserving mapping $f : S \rightarrow T$
- such that $f(o_S(s_1, \ldots, s_n)) = o_T(f(s_1), \ldots, f(s_n))$

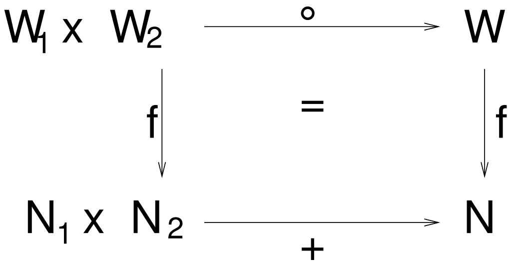


Example 1: Proportional Analogy



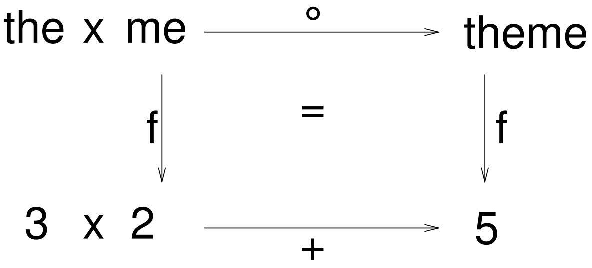
Example 2: Proportional Analogy

Relation between string concatenation and length of strings



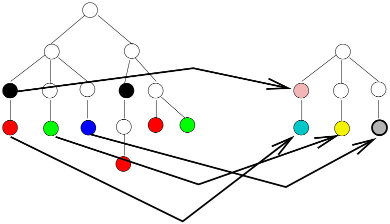
Example 2: Proportional Analogy

Relation between string concatenation and length of strings



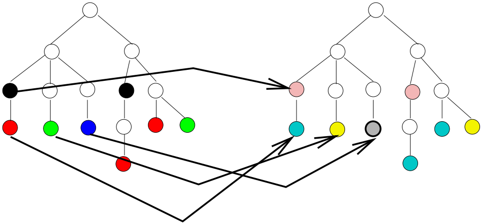
Transfer

- Based on mapping
- Carry-over of information from base to target
- „Inference“ of unknown characteristics of target
- Transfer/adaptation of a solution

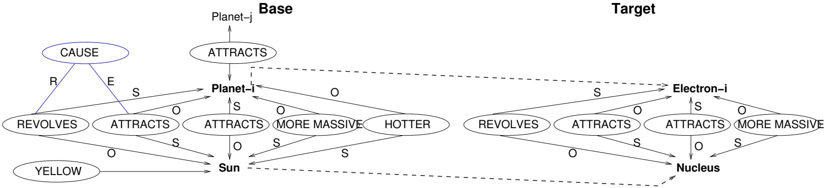


Transfer

- Based on mapping
- Carry-over of information from base to target
- „Inference“ of unknown characteristics of target
- Transfer/adaptation of a solution



Example – Transfer



Learning

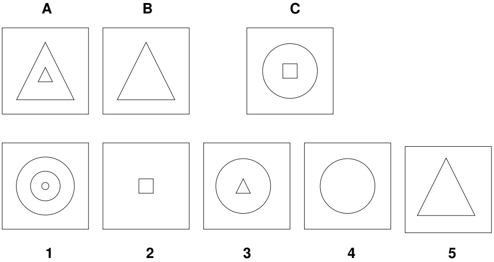
- Acquisition of more general schemes or rules by abstraction (Solar system, atom $\hookrightarrow$ central force system)
- Analogy as "beginner's strategy": Acquistion of general concepts makes analogical reasoning obsolete!

Computational Cognitive Science Systems

- Forbus, Falkenhainer & Gentner (1989): SME
  Naive physics
- Hummel & Holyoak (1997): LISA
  Problem solving (between domain)
- Anderson & Thompson (1989): modified ACT
  Programming/geometrical proofs (within domain)

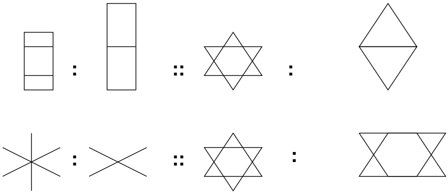
Cognitive AI Systems

- Evans (1968): Geometrical analogies (intelligence test)
- Veloso & Carbonell (1993): Plan construction (Prodigy)
- O'Hara (1992): geometrical analogies (PAN)
- Hofstadter (1995): letter strings (Copycat)



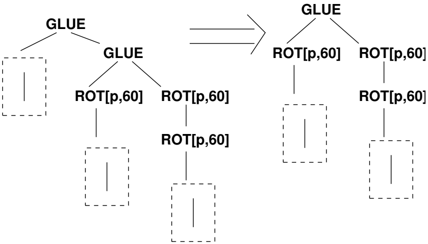
Cognitive AI Systems

- Evans (1968): Geometrical analogies (intelligence test)
- Veloso & Carbonell (1993): Plan construction (Prodigy)
- O'Hara (1992): geometrical analogies (PAN)
- Hofstadter (1995): letter strings (Copycat)



Cognitive AI Systems

- Evans (1968): Geometrical analogies (intelligence test)
- Veloso & Carbonell (1993): Plan construction (Prodigy)
- O'Hara (1992): geometrical analogies (PAN)
- Hofstadter (1995): letter strings (Copycat)



Cognitive AI Systems

- Evans (1968): Geometrical analogies (intelligence test)
- Veloso & Carbonell (1993): Plan construction (Prodigy)
- O'Hara (1992): geometrical analogies (PAN)
- Hofstadter (1995): letter strings (Copycat)

abc : abd :: kji : ??

Cognitive Plausibility of Approaches

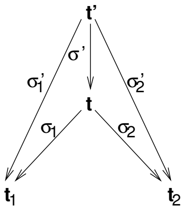
- Unflexible representation
  (additional re-representation mechanisms in PAN and Copycat)
- Mapping of graphs $\hat{=}$ subgraph isomorphism problem (NP-hard)
  (heuristics in SME and LISA)
- Transfer only „carry-over and replace“,
  no real adaptation (permutation, deletion)
- Learning not addressed
  or modelled by an additional mechanism (ACT)

Our Approach

- Anti-Unification (Reynolds, 1970): Mapping via common abstraction
- Mapping is governed directly by the common structure of base and target
- Allows use of equational theories for re-representation (Schmid, Gust, Kühnberger, Burghardt, 2003)
- Defined for first- and second-order case (Hasker, 1995)

AU Example

- $t_1 = \textit{attracts(sun, planet-i)}$
- $t_2 = \textit{attracts (nucleus, electron-i)}$
- can be generalized to
  $t = \textit{attracts(central-body, orbiter)}$
  where '*central-body*' and '*orbiter*' are variables
- Calculating the abstraction results in the mapping
  $\varphi = \{(\textit{sun},\textit{nucleus}) \mapsto \textit{central-body}$,
  $(\textit{planet-i},\textit{electron-i}) \mapsto \textit{orbiter}\}$
- Thereby, the necessary substitutions of variables by constants are also known:
  $\sigma_1 = \{\ \textit{central-body} \mapsto \textit{sun},\ \ \textit{orbiter} \mapsto \textit{planet-i}\ \}$ and
  $\sigma_2 = \{\ \textit{central-body} \mapsto \textit{nucleus},\ \ \textit{orbiter} \mapsto \textit{electron-i}\ \}$
- Now the mapping of *sun* onto *nucleus* can be performed by applying first φ and then σ_2.

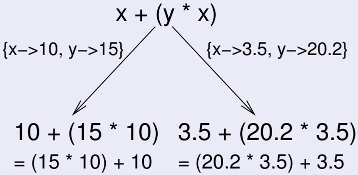


First-Order AU Modulo Equational-Theories

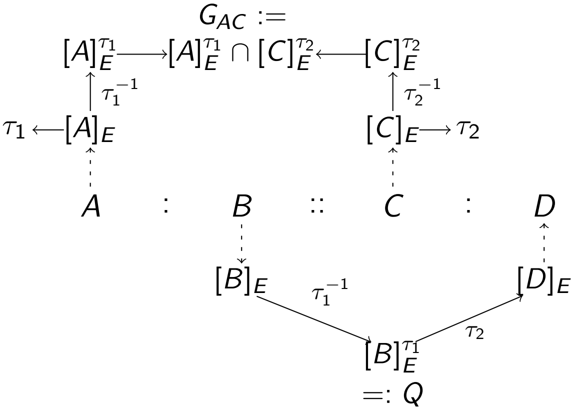
- Syntactic AU of $10 + (15 \cdot 10)$ and $(20.2 \cdot 3.5) + 3.5$ is $x + y$
- The fact, that each term is the addition of a constant and a product containing this constant got lost in generalization!
- Equational theories can be used to model knowledge about re-representations in a natural way

Simple Theory

$$x + 0 =_E x$$
$$x + y =_E y + x$$
$$x+(y+z) =_E (x+y)+z$$



Solving Proportional Analogies with E-Generalization



(BA Stephan Weller)

2nd Order Anti-Unification

- 1st order AU of $10 + (15 \cdot 10)$ and $3.5 - (20.2 \cdot 3.5)$ is x
- The fact, that each term is built by an operation over a constant and a product got lost in generalization!
- 2nd order AU can be used to model generalization over function/predicate symbols

- Result: $x\ F\ (y \cdot x)$
  where x and y are object variables and F is a function variable
- Applied to programming by analogy (Wagner and Schmid) in context of our work on inductive program synthesis techniques

Programming by Analogy

Fac-Problem: If the factorial of 3 is calculated as $3 \cdot 2 \cdot 1 \cdot 1$ what is the factorial for a natural number n?

Fac-Solution: $fac(n) = if(n{=}0,1,n \cdot fac(n\text{-}1))$.

NSum-Problem: If the neg. sum of 3 is calculated as $((0-1)-2)-3$ what is the neg. sum for a natural number n?

Programming by Analogy

Fac-Problem: If the factorial of 3 is calculated as $3 \cdot 2 \cdot 1 \cdot 1$ what is the factorial for a natural number n?

Fac-Solution: $fac(n) = if(n{=}0,1,n \cdot fac(n\text{-}1))$.

NSum-Problem: If the neg. sum of 3 is calculated as $((0-1)-2)-3$ what is the neg. sum for a natural number n?

Fac-Unfolding:

if $n = 0$ then 1
else if $n = 1$ then $1 \cdot 1$
else if $n = 2$ then $2 \cdot (1 \cdot 1)$
else if $n = 3$ then $3 \cdot (2 \cdot (1 \cdot 1))$

NSum-Problem:

if $n = 0$ then 0
else if $n = 1$ then $0 - 1$
else if $n = 2$ then $(0-1)-2$
else if $n = 3$ then $((0-1)-2)-3$

Programming by Analogy

Fac-Unfolding:
if $n = 0$ then 1
else if $n = 1$ then $1 \cdot 1$
else if $n = 2$ then $2 \cdot (1 \cdot 1)$
else if $n = 3$ then $3 \cdot (2 \cdot (1 \cdot 1))$

NSum-Problem:
if $n = 0$ then 0
else if $n = 1$ then $0 - 1$
else if $n = 2$ then $(0 - 1) - 2$
else if $n = 3$ then $((0 - 1) - 2) - 3$

Fac-NSum-Generalization:
if n=0 then **x**
else if n=1 then 1 F **x**
else if n=2 then 2 F (1 F **x**)
else if n=3 then 3 F (2 F (1 F **x**))

$\sigma_{\text{fac}} = \{x \mapsto 1, F \mapsto (\cdot\ \pi_1, \pi_2)\}$
$\sigma_{\text{nsum}} = \{x \mapsto 0, F \mapsto (-\ \pi_2, \pi_1)\}$
Note that arguments of subtraction op are reversed!

Programming by Analogy

Fac-NSum-Generalization:
if n=0 then **x**
else if n=1 then 1 *F* **x**
else if n=2 then 2 *F* (1 *F* **x**)
else if n=3 then 3 *F* (2 *F* (1 *F* **x**))

$\sigma_{\text{fac}} = \{x \mapsto 1, F \mapsto (\cdot\ \pi_1, \pi_2)\}$
$\sigma_{\text{nsum}} = \{x \mapsto 0, F \mapsto (-\ \pi_2, \pi_1)\}$
Note that arguments of subtraction op are reversed!

- The abstract term captures the role of 1 and 0 as neutral element and of $\cdot$ and $-$ as 'combination-operator' respectively.
- Obtaining the target solution: by applying the found substitutions to the recursive solution of the base problem
- Fac-Solution: *fac(n) = if(n=0,1,n · fac(n-1))*.
- NSum-Solution: *nsum(n) = if(n=0, 0, nsum(n-1) − n)*.

Evaluation

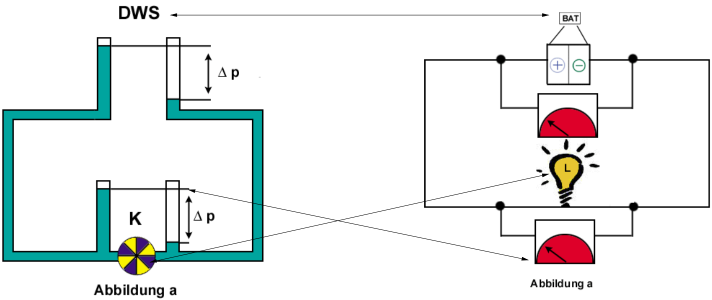
- AU is applicable to model all kinds of analogy
- By including equational theories, background knowledge can be included in a natural way
- Mapping via abstraction vs. direct mapping
  - Computational advantage
    (getting rid of subgraph isomorphism problem)
  - Cognitive plausibility
    (mapping guided by the common role of objects)
- Re-representation is modeled by taking into account all possible representations wrt a given theory simultaneously (investigation of re-representation triggers in human reasoning by Schmid and Jira)
- Learning by abstraction is a side-effect of analogical reasoning

Empirical Evidence

As much as needed vs. as much as possible

- Analogy by abstraction and SME: transfer of largest common substructure ("systematic") (vs. LISA, Copycat and others, "pragmatic")
- Analogy by abstraction: eager generalization during analogy making
- Assessing time of generalization (during or after mapping and transfer) is problematic
- Experiment by Eva Wiese (2007)
- Domain: Physics, water flow and current flow

Water Flow and Current Flow



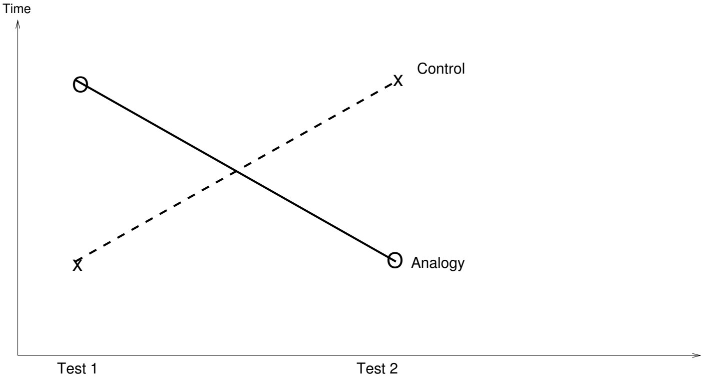
Abstraction of Ohm's Law?

- Tutorial: Water Flow (base analogy) or Current Flow (no analogy)
- Type-1 Problems: Amperage (I) and Voltage (V)
- Type-2 Problems: Amperage (I) and Resistance (R)
- Type-3 Problems: Interrelation of I, V, R; Ohm's Law

Design of the Study

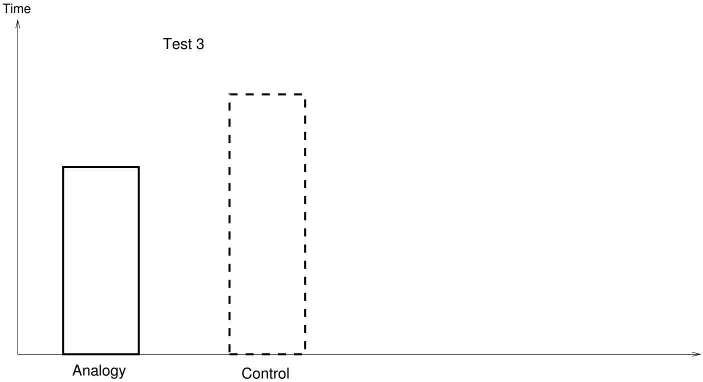
	E1	E2	K1	K2
Tutorial and Control Questions	Water		Current	
Test 1	Type-1	Type-2	Type-1	Type-2
Test 2	Type-2	Type-1	Type-2	Type-1
Test 3	Type-3	Type-3	Type-3	Type-3

Hypothesis: Transfer as much as possible



Analogical problem solving for I and V facilitates problem solving for I and R (likewise: I and R to I and V)

Hypothesis: Analogy triggers Generalisation



Analogical problem solving triggers generalization over structural relations (I, V, R)

Result Transfer

Mean time differences between Test 1 and Test 2 ($p < 0.001$); average solution time per problem type about 20 sec

	Type 1 (I, A)	Type 2 (I, V)
Experimental	-1,191	-0.979
Control	+1,385	+1,557

No speed-accurcy trade-off

- Average number of correct answers 88%
- Experimental group: 2% more wrong answers in Test 2
- Control group: 1% more wrong answers in Test 2
- No correlation between solution time and correct answers

Result Generalization

No systematic difference between Experimental and Control Group

- Experimental group about 1 sec slower than control group, about 10% more wrong answers
- Kind of problems used for Type 3 may be unsuitable
- New experiment: try to come up with a third domain

Application Potential

Analogy in Education

- Systematic support of acqusition of general principles/concepts by providing useful (structural similar but not identical) examples (Schmid, Wirth, Polkehn, 2003, Cognitive Science Quarterly)
- Support for understanding/discovering physical principles (Gust, Kühnberger; Wiese)

Application Potential

Analogy as *weak method* for AI systems

- Planning: search for a plan has exponential effort in the general case
  Alternative: Adaptation of existing plans
- Program synthesis: adaptation of existing programs

Final Remarks

- Analogy as an example for interdisciplinary research in higher cognition
- We investigate constraints and processes of human analogical reasoning
- We design "cognitive plausible" and formally sound algorithms
- Possible applications in many areas from education to AI technology