Educational Technologies
WS2008

Diagnosis and Feedback

Source: Erica Melis
Plan of the Course

28.10.2008 Introduction and overview
04.11.2008: Intelligent tutoring systems (1) - Cognitive Tutors
11.11.2008: Intelligent tutoring systems (2) – ActiveMath
18.11.2008: Student modelling
25.11.2008: Student modelling
02.12.2008: Pedagogical components, instructional planning
09.12.2008: Meta-cognitive support (1) – Help
16.12.2008: Error diagnosis and feedback
06.01.2009: Error diagnosis and feedback
13.01.2009: Collaborative learning technologies
20.01.2009: Multi-Media Learning principles
27.01.2009: Web-based systems
03.02.2009: Educational data mining
23.02.2009: Project presentations by students
Plan of Lecture

- Some cognitive foundations of feedback
- Diagnosis: generative approach
- Diagnosis: evaluative approach
- ActiveMath\' diagnosis and feedback
- Woz expriment
# Categories of Feedback

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
</table>
| Knowledge of performance [KP]         | • 15 of 20 task correct  
• 85% of the tasks correct           |
| Knowledge of result/response [KR]     | • correct / incorrect                                                      |
| Knowledge of the correct result [KCR] | • description of the correct response  
• indication of the correct response (e.g. for multiple choice questions) |
| Knowledge on task constraints [KTC]   | • Hints on type of task  
• Hints on task processing rules  
• Hints on subtasks  
• Hints on task requirements       |
| Knowledge about concepts [KC]         | • Hints on technical terms  
• Examples illustrating the concept  
• Hints on the conceptual context  
• Hints on concept attributes  
• Attribute-isolation examples     |
| Knowledge about mistakes [KM]         | • Number of mistakes  
• Location of mistakes  
• Type of errors  
• Sources of errors               |
| Knowledge on how to proceed [KH]      | • Bug-related hints for error correction  
• hints on task-specific strategies  
• Hints on task processing steps  
• Guiding questions  
• Worked-out examples              |
| Knowledge on meta-cognition [KMC]     | • Hints on meta-cognitive strategies  
• meta-cognitive guiding questions |

Source: Erica Melis
Educational Technologies WS 2008/09
Cognitive Feedback Models

- **Feedback and response certitude** [Kulhavy & Stock, 1989]
- **Feedback and mindful processing** [Bangert-Drowns et al, 1991]
- **Feedback and self-regulated learning** [Butler & Winne, 1995]
Feedback: some Cognitive Foundation [Narciss2006]

Learner factors
- cognitive, metacognitive
- motivational

Subjective representation of task requirements

Internal reference value

Internal controller
- Processes internal reference value and internal feedback
- Compares internal reference value and internal feedback
- Compares external feedback and internal reference value
- Compares internal and external feedback
- Generates internal control action

External feedback = external control action
Internal feedback of actual output

Control actuator
- Correcting
- Elaborating

Controlled variables
- Cognitive performance criteria
- Meta-cognitive performance criteria
- Motivational criteria

Internal sensor

External controller
- Processes external reference value and feedback
- Compares external reference value and feedback
- Generates external control action

External sensor

External representation of task requirements

Teacher - instructional medium

Instructional factors
- Instructional goals
- Instructional content and tasks

External reference value

Internal control action

Mastery of learning task requirements

Source: Erica Melis Educational Technologies WS 2008/09
Factors affecting the efficiency of external feedback

- **Requirements of learning tasks**

- **Internal loop factors**
  - learner’s knowledge, cognitive meta-cognitive and motivational strategies

- **External loop factors**
  - instructional goals, accuracy of diagnostic procedures, feedback design
Factors of Informative Feedback [Narciss 2004]

- **Instructional Context**
  - Learning Objectives
  - Learning Tasks
    - knowledge items
    - cognitive operations
    - meta-cognitive skills
  - Errors and Obstacles
    - typical errors
    - typical incorrect strategies
    - sources of errors

- **Individual Factors**
  - Learning objectives
  - Prior knowledge and skills
    - domain specific
    - meta-cognitive
  - Academic motivation
    - need of academic achievement
    - academic self efficacy
    - meta-motivational skills

- **Contents of feedback**
  - Evaluative component
  - Informative component
    - hints, cues
    - analogies
    - explanations
    - worked-out examples
    - etc.

- **Functions of Feedback**
  - Cognitive
  - Mete-Cognitive
  - Motivational

Source: Erica Melis   Educational Technologies WS 2008/09
Principles for designing (tutoring) feedback

► Factors determining the informative value of feedback
  ▶ Quality of feedback
  ▶ Characteristics of the instructional context.
  ▶ Cognitive and motivational characteristics of the learner

► Specifying the content of feedback elements
  ▶ Classification of feedback content.
  ▶ Cognitive task and error analyses.

► Selecting and specifying the form and mode of feedback presentation

Source: Erica Melis
Educational Technologies WS 2008/09
Diagnosis?

• **Technical Analysis** of student‘s problem solving
  - Is step/solution correct?
  - Where in solution graph?
  - Is step relevant?
  - What is wrong?
  - What is missing?

▶ **Cognitive diagnosis** of student‘s performance
  ▶ What has been understood by the student?
  ▶ What misconceptions can be observed?
  ▶ What is student‘s motivation? ...
ACT-R Theory of Cognition

ACT-R = adaptive control of thought [Anderson]
- Declarative memory
- Procedural memory

Learning happens in several phases:
- Learning declarative knowledge
- Turning declarative knowledge into procedural knowledge (goal directed), more efficient to use
- Strengthening procedural knowledge
ACT-R Theory of Cognition

Fundamental assumption: Cognitive skills are realized by procedural knowledge represented by production rules

=> Learner supposed to learn production rules of a domain
Production Rules

IF goal is to compute $x/d_1+y/d_2$ AND $d_1$ and $d_2$ are equal
THEN result is $(x+y)/(d_1+d_2)$

IF goal is to compute $x/d_1+y/d_2$ AND $d_1$ and $d_2$ are not equal
THEN set as subgoals to
- first extend $x/d_1$ and $y/d_2$ to the smallest common multiple of $d_1$ and $d_2$ and
- compute then the sum of the resulting fractions
ACT-R Model

- **ACT-R Cognitive Model:** A system that can solve problems in the various ways students can

- **Assumption:** declarative knowledge through instruction. Then converted and reorganized to procedures.

**Strategy 1:** IF the goal is to solve $a(bx+c) = d$

Then rewrite this as $abx + ac = d$

**Strategy 2:** IF the goal is to solve $a(bx+c) = d$

Then rewrite this as $bx + c = d/a$

**Misconception:** IF the goal is to solve $a(bx+c) = d$

Then rewrite this as $abx + c = d$
Diagnosis: Generative Approach

General Idea:

- Domain reasoner with relevant knowledge available to dynamically generate solutions
- Observe progress of learner and match input with steps/solutions generated by the domain reasoner
- Generate/choose feedback to student‘s input accordingly
Cognitive Tutor Technology: Use ACT-R theory to individualize instruction

**Cognitive Model:** A system that can solve problems in the various ways students can

- If goal is solve \(a(bx+c) = d\) Then rewrite as \(abx + ac = d\)
- If goal is solve \(a(bx+c) = d\) Then rewrite as \(abx + c = d\)
- If goal is solve \(a(bx+c) = d\) Then rewrite as \(bx + c = d/a\)

\[3(2x - 5) = 9\]
\[6x - 15 = 9\]
\[2x - 5 = 3\]
\[6x - 5 = 9\]

- **Model Tracing (diagnosis)** Follows student through their individual approach to a problem
  -> context-sensitive instruction (feedback)
Cognitive Tutor Diagnosis:
Use ACT-R theory to individualize instruction

Cognitive Model: A system that can solve problems in the various ways students can

IF goal is solve $a(bx+c) = d$
THEN rewrite as $abx + ac = d$

Hint message: "Distribute a across the parentheses."

Known? = 85% chance

$6x - 15 = 9$

$3(2x - 5) = 9$

$2x - 5 = 3$

$6x - 5 = 9$

IF goal is solve $a(bx+c) = d$
THEN rewrite as $abx + c = d$

Bug message: "You need to multiply c by a also."

Known? = 45%

• Model Tracing: Follows student through their individual approach to a problem -> context-sensitive instruction
Production Rules

IF goal is to compute x/d1 + y/d2 AND d1 and d2 are equal
THEN result is (x+y)/(d1+d2)

IF goal is to compute x/d1 + y/d2 AND d1 and d2 are not equal
THEN set as subgoals to
  - first extend x/d1 and y/d2 to the smallest common multiple of d1 and d2 and
  - to compute then the sum of the resulting fractions
Exercise, example

Compute: $\frac{5}{6} + \frac{4}{3}$

Answer: $\frac{5}{6} + \frac{4}{3} = \frac{5}{6} + \frac{8}{6} = \frac{13}{6}$

Correct?

Domain reasoner evaluates rules ... ... and succeeds to match all student steps
Exercise, example

Compute: $\frac{5}{6} + \frac{4}{3}$

Answer: $\frac{5}{6} + \frac{4}{3} = \frac{13}{6}$

Correct?

Domain reasoner evaluates rules ...

... and succeeds to match (BIG) student step
Exercise, example

Compute: $5/6 + 4/3$

Answer: $5/6 + 4/3 = 9/9$

Correct?

Domain reasoner evaluates rules ...

... and fails to match student steps

What´s the failure here?
Buggy Rules

- **Idea**: Extend domain reasoner such that it can recognize typical ´´buggy solutions´´

- **Introduce buggy rules:**

  IF goal is to compute $x/d1+y/d2$ AND $d1$ and $d2$ are not equal

  THEN result is $(x+y)/(d1+d2)$
Exercise

Compute: \( \frac{5}{6} + \frac{4}{3} \)

Answer: \( \frac{5}{6} + \frac{4}{3} = \frac{9}{9} = 1 \)

Correct?

Domain reasoner evaluates rules ...

... and succeeds to match student steps with buggy rule

=> feedback
Cognitive Tutors

Tutors following this approach have been developed at CMU for:

- College level physics
- High school algebra
- Geometry
- Lisp and Prolog programming
Generative Approach: General Problems

- Domain Reasoner is required for solution construction
- For rule based domain reasoner: Complete domain knowledge has to be encoded in rules

⇒ Difficult and laborious
⇒ For some domains unfeasable
Cognitive Tutors: Feedback Generation

- **Idea: Keep student on a correct solution path**
  - Analyse each student step and inform about errors immediately

- **2 possible error situations:**
  - Domain reasoner matches buggy student step with buggy rule
  - Domain reasoner fails to match student step with a rule execution sequence (i.e., fails to find buggy rule)
Tutor: Hello. Jane is “j“ years old. Peg is “p“ years old. Mary is Peg´s age minus Tom´s age. Tom is 3 years yonger than Jane. Peg is “x“ years older than Mark. Write an expression for Mary´s age.

Correct answer: p-(j-3)
Error Feedback in Ms. Lindquist

**Student:** (p-3)-(3-j)+(p+x)

**Tutor:** Hmm, no. Let me ask you an easier question. Use j and 3 to calculate Tom’s age.

⇒ “Hmm” indicates that the tutor fails to recognize the error
⇒ Tutor guides the student towards the subgoals of the current goal
Error Feedback in Ms. Lindquist

Student: 3-j
Tutor: No. Which is larger, the ´3´ or the ´j´? Please try again?

=> Tutor matches a buggy rule about confusion of ´smaller´ (i.e., yonger) and ´-´

⇒ Feedback associated with the buggy rule is provided to the student

⇒ Repeated errors result in more detailed feedback
  (e.g., some alternatives or solution parts)
Error Feedback in the Lisp Tutor

Define a function called ends that takes one argument, which must be a list, and returns a new list containing the first and last items in the argument. For example,

```
(ends '(a b c d)) = (a d)
```

CODE for ends

```
(defun <FUNCTION> <PARAMETERS>
  <PROCESS>)
```
Error Feedback in the Lisp Tutor

You will need to call the function CAR, but not yet. You need to construct a list containing the first item in the argument and the last item in the argument, so you need to call a list combining function here.

```lisp
(defun ends (lis)
    (car)
)
```
Error Feedback in the Lisp Tutor

Remediation based on goal tree of problem
Evaluative Approach, Constrain-Based Diagnosis

General idea:

- Forget about solution, solution construction and problem solving process
- Analyse a solution state by constraints a correct solution has to satisfy
Example Constraint

Cr: \((x+y)/d\) is given as the answer to \(x/d_1 + y/d_2\)
Cs: \(d=d_1=d_2\)

Constraint = \(<Cr,Cs>\)
Cr=relevance condition
Cs=satisfaction condition

Student inputs solution
=> Constraint engine computes relevant constraints
=> Constraint engine evaluates relevant constraints
to detect unsatisfied conditions
Exercise

Compute: $\frac{5}{6} + \frac{4}{3}$

Answer: $\frac{5}{6} + \frac{4}{3} = \frac{13}{6}$

Correct?

Constraint engine evaluates constraints ...  
... and detects no unsatisfied relevant constraints
Exercise

Compute: 5/6 + 4/3
Answer: 5/6 + 4/3 = 9/9 = 1
Correct?

Constraint engine evaluates constraints ...
... and detects unsatisfied relevant constraint

=> Suitable feedback issued
Constraint-Based Tutors [Ohlsson, Mitrovich]

- SQL database commands
- Punctuation
- Database modelling
Underlying Learning Theory

- Humans make mistakes because the declarative knowledge they learn is not turned into appropriate procedural knowledge.

- By catching themselves or being caught by a tutor making mistakes, students modify our procedural knowledge.
Underlying Learning Theory

⇒ Fundamental assumption: learning happens by error recognition and error correction

[Ohlsson]

Analysis: by constraint evaluator

Explanation: learner makes errors / does not master corresponding procedural knowledge
User Modelling in Constraint-Based diagnosis

User Model consists of:
• Constraints labeled with number of violations

User Model is updated by:

▶ Analysis of the constraints violated by the learner when solving exercises

User Model is used to:
• Exercises selection: select exercises for which often violated constraints are relevant

Source: Erica Melis  Educational Technologies WS 2008/09
Feedback Generation

- Feedback is directly associated with constraints
- Student’s solution is evaluated only when student finishes exercise or requests evaluation

⇒ Analysis and feedback generation are delayed
Think of example for constraint-based diagnosis/feedback
Feedback in SQL Tutor

- Feedback is directly associated with constraints
- Different levels of detail from right/wrong up to solution parts

Feedback Strategy:
- When several constraints are violated provide feedback for one only (simpler to cope with)
- First give less information, repeated errors result in more information
### Feedback of the SQL Tutor

**Example 1:** For each director, list the director's number and the total number of awards won by comedies he/she directed if that number is greater than 1.

<table>
<thead>
<tr>
<th>Correct solution:</th>
<th>Student's solution:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT DIRECTOR, SUM(AA Won) FROM MOVIE WHERE TYPE='comedy' GROUP BY DIRECTOR</td>
<td>SELECT DIRECTOR, SUM(AA Won) FROM DIRECTOR JOIN MOVIE ON DIRECTOR=DIRECTOR.NUMBER WHERE TYPE='comedy'</td>
</tr>
<tr>
<td>HAVING SUM(AA Won)&gt;1</td>
<td></td>
</tr>
</tbody>
</table>

**SQL-Tutor:**
- You do not need all the tables you specified in `FROM`!
- You need to specify the `GROUP BY` clause! The problem requires summary information.
- Specify the `HAVING` clause as well! Not all groups produced by the `GROUP BY` clause are relevant in this problem.
- If there are aggregate functions in the `SELECT` clause, and the `GROUP BY` clause is empty, then `SELECT` must consists of aggregate functions only.
Solutions?

- Evaluative approach requires no solution construction
- However, SQL Tutor stores one solution for each exercise:
  - Some constraints match in their conditions against solution (e.g., to detect missing parts)
  - Solution parts can be provided as feedback
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>(rule-based) Generative Approach</th>
<th>(constraint-based) Evaluative Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Knowledge Representation</strong></td>
<td>Production Rules</td>
<td>Constraints</td>
</tr>
<tr>
<td><strong>What is Evaluated?</strong></td>
<td>Action/Progress</td>
<td>Solution State</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Immediate</td>
<td>Delayed</td>
</tr>
<tr>
<td><strong>Diagnosis if No match</strong></td>
<td>Incorrect</td>
<td>Correct</td>
</tr>
<tr>
<td><strong>Problem Solved</strong></td>
<td>´Done´ Productions</td>
<td>No violated constraints</td>
</tr>
</tbody>
</table>
Comparison: Implementation Effort

- Experiment implementing the same tutor in both approaches: 43 constraints vs. 76 production rules
  ⇒ more implementation effort of generative approach

• Explanation:
  • Constraints correspond to basic rules
  • Rules to decompose goals and construct solution are extra effort

⇒ Advantages constraint-based evaluative approach
Comparison: Feedback

- Both can provide feedback to errors
- In both (some) feedback is (more or less) directly attached to buggy rules or constraints
- In both quality of error feedback considerably depends on quality and detail of rules/constraints

• Rule-based generative approach can additionally provide:
  – goal decomposition hints
  – follow-up steps wrt. solution path
  – strategic hints
Comparison: Feedback Problems

- Problem rule-based generative approach:
  Unexpected situations are treated as error

- Problem constraints-based evaluative approach:
  Error confusion when many errors detected
Complexity of goal structure:

For problems with complex and deep goal structures
⇒ Rule-based generative approach provides better feedback
⇒ But development of domain reasoner may be very complex
When to Choose which Approach?

Information Richness of Solution:

Does solution state provide sufficient information to be analysed by constraint-based evaluative approach? (extrem case: solution is yes/no)

=> Constraint-based evaluative approach is not suitable for problems whose solutions provide no information
Overview

- Knowledge representation
- Diagnosis: evaluative approach, semantic evaluation
- Diagnosis: generative approach, domain reasoner
- Feedback: tutorial and presentation strategies
Exercise Subsystem Architecture

Source: Erica Melis  Educational Technologies WS 2008/09
Knowledge Representation of Exercises

- **Finite state machine of interactions**
- **Types of nodes**
  - Static: task description, feedback
  - Interactive: user interaction
- **Types of Transitions**
  - Conditional transition triggered by user answer
  - Conditional transition triggered by user request
  - Unconditional transition
Finite State Machine KR

Hier BILD und formale repr,
Why this Representation?

◧ Suitable for many domains
  ◦ Generic multiple choice exercises
  ◦ Semantic representation for free user answers allows for deep diagnosis

◧ Representation suitable for automatic generation
  ◦ Proper separation of low level building blocks

◧ Declarative representation suitable for authoring
  ◦ Automated authoring tools supported

◧ Reusable exercise representation
  ◦ Exercise solution space can be traversed in many ways
Sample solution space of an exercise
Diagnosis

- Semantic Evaluation with Computer Algebra System
- Randomized solution spaces using CAS
- Domain Reasoner support possible
  - Refined diagnosis upon user actions
  - Stepwise solution guiding strategies possible
  - Typical error matching helps revealing misconceptions
Semantic Evaluation in different contexts

Exercise

The total average slope of a curve

Compute the average slope of the depicted curve between $A = (x_{A1}, y_{A1}) = (0, 100)$ and $\theta = (x_{A2}, y_{A2}) = (4000, 200)$.

$m_{avg} = \frac{\text{rise}}{\text{run}}$

Not quite. Please check the formula for the average slope. It seems that you mixed numerator and denominator.

$m_{avg} = \frac{100}{4000}$

Well, this fraction can be further simplified.

$m_{avg} = \frac{1}{40}$

Well done!
Diagnosis Broker Architecture
Diagnosis: Generic Queries

Query consists of

- **name** of the action (e.g. getNextStep, evaluate)
- **context** of the action identifying the set of applicable semantic rules (e.g. arithmetic, differentiation, logic)
- **body** (e.g. task, user answer, correct answer)
- **number** of iteration (defines the granularity of a reasoning step)
Diagnosis: Prolog Domain Reasoner [Zinn2006]

- Analysis of human-human dialogues
- Formalization of reasoning in symbolic differentiation
- operationalization
Dialogue 1

T-1  Try the following one: \( \frac{5}{\sqrt{x^6 + 2x}} \)

S-2a  \( 5(x^6 + 2x)^{-\frac{1}{2}} \)

S-2b  \( = -\frac{5}{2}(x^6 + 2x)^{-\frac{3}{2}}(6x^5 + 2) \)

S-2c  \( = -5(x^6 + 2x)^{-\frac{3}{2}}(3x^5 + 1) \)

S-2d  i think thats right ut im not too sure

T-3a  That’s very good. You really have got to grips with the chain rule for algebraic expressions.
Dialogue 2

T-3b Let’s move on to other functions. Try to differentiate $\log(x^2 + 6x - 1)$.

S-4 I’m not sure what you get when you differentiate log. It is $e^x$?

T-5 No. I’ll tell you: $\frac{d}{dx} \log(x) = \frac{1}{x}$

S-6 I’m still not sure how to do this one

T-7 First you need to identify the functions in the composition $(f(g(x)))$. By the way, you really need to remember what the derivative of log is...

S-8 I still don’t know what to do

T-9a Think about the example that you read in the beginning.

T-9b Try to identify z again and then y as a function of z.

S-10 $z = x^2 + 6x - 1$ I’m not sure about what y is

T-11 That’s good so far. Now think where z appears in the expression

S-12 $y = \log z$

T-13 Yes. That’s right. Now can you put it all together?

S-14 $\frac{1}{x^2 + 6x - 1} (2x + 6)$

T-15a Yes. That’s it. We could write that as $\frac{2(x + 3)}{x^2 + 6x - 1}$
Dialogue 3

T-15b Now let’s try one with trig functions. Try \( \frac{1}{\sin^3 x} \)
Remember that the derivative of \( \sin \) is \( \cos \)

S-16 \((\sin^3 x)^{-1} - (\sin^3 x)(3\cos^2 x)\)

T-17 Think of \( \sin^3(x) \) as \((\sin(x))^3\)

S-18 \(3(\sin(x))^2(\cos x)\)

T-19 That’s much better. Now can you solve the original problem (which is a little different)?

S-20 \((\sin(x))^{-3} = -3(\sin(x))^{-2}(\cos x)\)

T-21 Almost. Remember that the derivative of \( x^{-n} \) is \(-nx^{-n-1}\).

T-22 I think you know the answer: \(-\frac{3\cos(x)}{\sin^4 x}\)

S-23 yes that is what i was thinking [...]
### Expert Rules of Symbolic Differentiation

<table>
<thead>
<tr>
<th>Rule</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sine Rule</td>
<td>( \frac{d}{dx} \sin(x) = \cos(x) )</td>
</tr>
<tr>
<td>Logarithmic Rule</td>
<td>( \frac{d}{dx} \log(x) = \frac{1}{x} )</td>
</tr>
<tr>
<td>Power Rule</td>
<td>( \frac{d}{dx} [x^n] = n \cdot x^{n-1} )</td>
</tr>
<tr>
<td>Constant Multiple Rule</td>
<td>( \frac{d}{dx} [c \cdot f(x)] = c \cdot \frac{d}{dx} [f(x)] )</td>
</tr>
<tr>
<td>Sum Rule</td>
<td>( \frac{d}{dx} [f(x) \pm g(x)] = \frac{d}{dx} [f(x)] \pm \frac{d}{dx} [g(x)] )</td>
</tr>
<tr>
<td>Chain Rule For Power Functions</td>
<td>( \frac{d}{dx} [(f(x))^n] = n \cdot [f(x)]^{n-1} \cdot \frac{d}{dx} [f(x)] )</td>
</tr>
<tr>
<td>General Chain Rule</td>
<td>( \frac{d}{dx} [f(g(x))] = \frac{d}{dg} [f(g(x))] \cdot \frac{d}{dx} [g(x)] )</td>
</tr>
</tbody>
</table>

\( c \) is any real number, \( f(x) \) and \( g(x) \) are any functions
Encoding of Expert Rules

- \( \frac{d}{dx} \sin(x) = \cos(x) \)
- derive(Var, sin(Var), cos(Var)).
- the query
  \(?-\text{derive}(x, \sin(x), \text{Answer}).\)
  yields Answer = \cos(x).
- \( \frac{d}{dx} [x^n] = n \cdot x^{n-1} \)
- derive(Var, Var^N, N*(Var^N1)) :- freeof(Var, N), N1 is N-1.
- \( \frac{d}{dx} [f(x) + g(x)] = \frac{d}{dx} [f(x)] + \frac{d}{dx} [g(x)] \)
- derive(Var, A+B, A1+B1) :- derive(Var, A, A1), derive(Var, B, B1).
Expert Rules contd.

\[ \frac{d}{dx} ([f(x)]^n) = n \cdot [f(x)]^{n-1} \cdot \frac{d}{dx} [f(x)] \]

\[ \text{derive(Var, } X^N, (N*(X^{N1}))*XP) \text{ } \text{:- not (Var = X),} \]
\[ \text{N1 is N - 1,} \]
\[ \text{derive(Var, X, XP).} \]

the query

\[ \text{?- derive(x, 5*(x^6+2*x)^(-0.5), Answer).} \]

yields \( \text{Answer} = 5*(-0.5*(x^6-2*x)^(-1.5)*(6*x^5+2*1)) \).

\[ \text{Prolog returns results only, no intermediate steps} \]
Expert Rules extended

\[
\text{derive}(X, \sin(X), \cos(X), [\sinRule(\sin(X) = \cos(X))]).
\]

\[
\text{derive}(X, X^N, (N*(X^N1)), [\text{powerRule}(\text{deriv}(X^N)=(N*(X^N1)))]):-
\]

\[
\text{freeof}(X, N),
\]

\[
N1 \text{ is } N - 1.
\]

\[
\text{derive}(X, A+B, A1+B1, [\text{sumRule}(\text{deriv}(A+B)=\text{deriv}(A)+\text{deriv}(B)),\text{Rules}]):-
\]

\[
\text{derive}(X, A, A1, R1),
\]

\[
\text{derive}(X, B, B1, R2),
\]

\[
\text{append}(R1, R2, \text{Rules}).
\]

Now, the query

\[
?-\text{derive}(x, 5*(x^6+2*x)^(-0.5), \text{Answer}, \text{ Explain}).
\]

returns same Answer, and Explain contains solution graph that shows each of rule used to compute Answer:
Solution Graph in Prolog

\[
\text{[const_mult_rule( = (deriv(x, 5*(x^6+2*x)^number(-, [1], [2])))}
\]
\[
\text{5*deriv(x, (x^6+2*x)^number(-, [1], [2]))])}
\]
\[
\text{[chain_rule_pow( = (deriv(x, (x^6+2*x)^number(-, [1], [2])))}
\]
\[
\text{number(-, [1], [2])*(x^6+2*x)^number(-, [3], [2])*deriv(x, x^6+2*x))}
\]
\[
\text{[sum_rule( = ( deriv(x, x^6+2*x))}
\]
\[
\text{deriv(x, x^6)+deriv(x, 2*x)))],}
\]
\[
\text{[power_rule( = ( deriv(x, x^6))}
\]
\[
\text{6*x^5)])],}
\]
\[
\text{[constant_multiple_rule( = ( deriv(x, 2*x))}
\]
\[
\text{2*deriv(x, x)))],}
\]
\[
\text{[linear_x( = ( deriv(x, x))}
\]
\[
\text{1))]]]]]]
\]
# Buggy Rules in Symbolic Differentiation

<table>
<thead>
<tr>
<th>Missing Inner Layer (Chain Rule)</th>
<th>[ \frac{d}{dx} (x^3 - 3x)^5 \not\equiv 5(x^3 - 3x)^4 ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrong Exponent (Chain/Power Rule)</td>
<td>[ \frac{d}{dx} (5x^3 - 6)^{-3} \not\equiv -3(5x^3 - 6)^{-2}(15x^2) ]</td>
</tr>
<tr>
<td>Missing Exponent (Chain/Power Rule)</td>
<td>[ \frac{d}{dx} (x^3 - 3x)^5 \not\equiv 5(x^3 - 3x)(3x^2 - 3) ]</td>
</tr>
<tr>
<td>Incorrect Basic Rule (Sine Rule)</td>
<td>[ \frac{d}{dx} \sin(x) \not\equiv -\cos(x) ]</td>
</tr>
<tr>
<td>Missing Bracketing (Op. Precedence)</td>
<td>[ \frac{d}{dx} (x^3 - 3x)^5 \not\equiv 5(x^3 - 3x)^4 3x^2 - 3 ]</td>
</tr>
<tr>
<td>Erroneous Transformation (Rewriting)</td>
<td>[ \frac{1}{(5x^3 - 6)^3} \not\equiv (5x^3 - 6)^{-\frac{1}{3}} ]</td>
</tr>
</tbody>
</table>
Expert Power Rule and a Buggy Version

derive(\texttt{expert},
    X,
    X^N,
    N*(X^N1),
    [\texttt{powRule(deriv(X^N)=N*(X^N1))}] :-
    \texttt{freeof(X, N)}, N1 is N - 1.

derive(\texttt{buggy},
    X,
    X^N,
    N*(X^N1),
    [\texttt{powRule(deriv(X^N)=N*(X^N1)), wrongPow(N1)}] :-
    N < 0,
    \texttt{freeof(X, N)},
    N1 is N + 1.
Example Queries

?- derive(RuleType, x, x^(-4), (-4*(x^(-5))), Explain)
=> RuleType = expert
    Explain = [powerRule(deriv(x^(-4))=(-4*(x^(-5))))]

?- derive(RuleType, x, x^(-4), (-4*(x^(-3))), Explain)
=> RuleType = buggy
    Explain = [powerRule(deriv(x^(-4))=(-4(x^(-3))), wrongPow(-3)]

?- findall( (RuleType,StudentAnswer),
           derive(RuleType, x, x^(-4), StudentAnswer, Explain),
           AllDiagnoses )

=> AllDiagnoses = "<all recognisable answers>"
Main services of Domain Reasoner

① Define (differentiation) tasks
   • solves given task
   • computes solution
   • constructs solution graph

② Get Solution
③ Get Next Step
④ Get Diagnosis
Feedback Generation based on DR

- works with annotations on solution graph
- if learners’ problem solving step *matches expert node*, then *mark* this step as visited, and provide appropriate *feedback*
- if learner’s problem solving step does not match expert node, then generate alternative buggy solution graphs to search for matching buggy node
  - supporting diagnosis of multiple bugs
- if learner requests help, then search graph for *next unvisited node*, and generate *feedback* exploiting node’s content.
- generation of *next step help* flexible
  - nodes have *specificity counter*
Example of intelligent automatic help

Enter the function to differentiate:

\[ \sin x^3 \]

Okay, let us differentiate this:

\[ \cos x^3 \]

This is incorrect.

Try again:

You have to apply the following rule: chain rule

Let the function \( f \) be differentiable on the interval \( I \) and let \( g \) be differentiable on the interval \( f(I) \). Then also the composed function \( h = g \circ f \) with \( h(x) = g(f(x)) \) is differentiable on \( I \) and we have:

\[ h'(x) = g'(f(x)) \cdot f'(x) \] for all \( x \in I \).

The factor \( g'(f(x)) \) is called the \textbf{outer derivative}, the factor \( f'(x) \) is called the \textbf{inner derivative}.
Feedback: Tutorial and Presentation Strategies

- Exercises can be used for different educational goals
- Educationalists and cognitive psychologists design the feedback
- In ActiveMath different strategies can be applied to same exercises
  - Adapting strategies to the learner’s context
  - Reusing the same exercise for several purposes
  - Experimental platform for creating and evaluating strategies
Feedback: Tutorial and Presentation Strategies
Presentation Strategies

$m_{AB} = 40$.

Not quite. Please check the formula; you mixed numerator and denominator.

$m_{AB} = \frac{1}{40}$.

Well done!

$m_{AB} = 40 \text{ 😞}$.

Not quite. Please check the formula for the average slope. It seems that you mixed numerator and denominator.

$m_{AB} = \frac{1}{40} \text{ 😊}$.

Well done!
Tutorial and Presentation Strategies

Löse die Aufgabe und klicke auf "Weiter" wenn du fertig bist:

\[
\frac{3}{11} + \frac{5}{11} = \frac{3+5}{11} = \frac{8}{11}
\]

Endergebnis = \(\frac{8}{11}\)

Löse die Aufgabe und klicke auf "Weiter" wenn du fertig bist:

\[
\frac{3}{11} + \frac{5}{11} = \frac{3+5}{11} = \frac{8}{11} \text{ Richtig!}
\]

Endergebnis = \(\frac{8}{11}\) Richtig!

Die Übung ist beendet, bitte das Fenster schließen.

Source: Erica Melis  Educational Technologies WS 2008/09
Delayed Feedback Strategies

![Image of ActiveMath screenshot]

Source: Erica Melis  Educational Technologies WS 2008/09
Erroneous Example Strategy

Klaus sollte $\frac{1}{3} + \frac{2}{7}$ berechnen und das Ergebnis wenn nötig kürzen bzw. umwandeln. Er ging folgendermaßen vor:

- **Hauptnener**
  - Der Hauptnener von $\frac{3}{5}$ und $\frac{2}{7}$ ist 56

- **Addieren**
  - $\frac{1 \cdot 7}{5 \cdot 7} = \frac{7}{56}$

- **Kürzen**
  - $\frac{7}{56}$ gekürzt mit 7 ergibt $\frac{1}{8}$

Markiere die Stelle, an der Klaus ein Fehler passiert ist!

Wie hätte Klaus richtig vorgehen müssen? Setze seine begonnene Rechnung mit dem richtigen Schritt im Arbeitsfeld fort!
Modelling Exercise Strategy

Petras Mutter hat eingekauft: $\frac{3}{8}$ kg Wurst, $\frac{3}{4}$ kg Fleisch und $\frac{1}{2}$ kg Kaffee.

Wie viel kg wiegen diese Sachen in ihrer Einkaufstasche?

Schreibe die Aufgabenstellung ohne Maßangaben ins Aufgabenfeld und gib die Lösung im Ergebnisfeld an!

Aufgabenfeld:

Ergebnisfeld:

Hier können Sie Ihre Lösung schrittweise machen:

1. Schritt Erweitern

$\frac{3}{8}$ erweitert mit $2$ ergibt $\frac{6}{8}$

2. Schritt Nur Formeleingabe
ATuF Interface

Übung zum Addieren von ungleichnamigen Brüchen

Petras Mutter hat eingekauft: \(\frac{3}{5}\) kg Wurst, \(\frac{3}{4}\) kg Fleisch und \(\frac{1}{2}\) kg Kaffee.

Wie viel kg wiegen diese Sachen in ihrer Einkaufstasche?

Schreibe die Aufgabenstellung ohne Maßangaben ins Aufgabenfeld und gib die Lösung im Ergebnisfeld an!

Aufgabenfeld:

\[
\begin{array}{c}
3 \\
8 \\
\end{array}
\quad +
\begin{array}{c}
3 \\
4 \\
\end{array}
\quad -
\begin{array}{c}
1 \\
2 \\
\end{array}
= \begin{array}{c}
\quad \frac{15}{8}
\end{array}
\]

Ergebnisfeld:

\[
\frac{15}{8}
\]

Die Aufgabe stimmt, aber die eingegebene Ergebnis ist leider falsch. Versuch die Aufgabe schrittweise zu lösen.
## Self-Regulation Strategy

<table>
<thead>
<tr>
<th>Durchschnittliche Fahrtgeschwindigkeit:</th>
<th>en:</th>
<th>te</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Landstraße</strong> 75 km/h</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Autobahn</strong> 120 km/h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zrückgelegte Strecke:</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Landstraße</strong> 12.5 km</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Autobahn</strong> 10 km</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fahrtdauer zu den Tankstellen:</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Gasolina</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Fuel</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unterschied in der Fahrtdauer:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zu der Tankstelle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fährt man weniger, als zu der Tankstelle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Bestimmung der Fahrtdauer zu den Tankstellen:**

Eine Proportionalitätsaufgabe

<table>
<thead>
<tr>
<th>Gasolina</th>
<th>Fuel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unterschied in der Fahrtdauer: Zu der Tankstelle</th>
<th>en:</th>
<th>te</th>
</tr>
</thead>
<tbody>
<tr>
<td>fährt man weniger, als zu der Tankstelle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Self-Regulation Strategy
Self-Regulation and Reflection Phases

Schau Dir an, welche Reihenfolge der Schritte Du gewählt hastest, um diese Teilfrage zu bearbeiten.

Restvolumen des Tanks
Benzinpreis pro Liter an den Tankstellen: Gasolina / Fuel
Benzinpreis für das Volltanken an den Tankstellen: Gasolina / Fuel
Unterschied im Benzinpreis für das Volltanken
Benzinpreis pro Liter an den Tankstellen: Gasolina / Fuel

Beschreibe Dein Vorgehen: [ ]

Source: Erica Melis Educational Technologies WS 2008/09
WoZ

- Design your own (simple) exercise(s)
- Decide for one diagnosis approach and prepare rules/constraints for your tutor (System + Observer)
- Test your approach with the Learner
- What happens?
  - Is your feedback suitable?
  - Is your learner missing hints, suggestions?
  - ...

Source: Erica Melis
Educational Technologies WS 2008/09