

Eine kurze Einführung in ISAC

[Wie starte ich ISAC ?](#)

[Wie wähle ich ein Beispiel und starte eine Rechnung ?](#)

[Wie lasse ich ISAC Schritt für Schritt vorrechnen?](#)

[Wie kann ich selbst Zwischenschritte zur Rechnung eingeben ?](#)

[Welche Begründungen gibt es für einen Rechenschritt ?](#)

[Die Taktik, die den Schritt erzeugt hat.](#)

[Die weiteren Zwischenschritte einer 'Rewrite'-Taktik.](#)

[Alles Mathematik-Wissen im Kontext.](#)

[Woher weiß ISAC seine Taktiken ?](#)

[Woher weiß ISAC, aus welcher Methode die Taktiken zu nehmen sind ?](#)

[Kann ich selbst neue Rechnungen eingeben ?](#)

ISAC ist eine experimentelle Software, die Aufgaben der angewandten Mathematik lösen kann.

Lernen kannst Du von ISAC, indem Du die Zwischenschritte verfolgst, in denen ISAC eine Aufgabe löst *und* indem Du selbst Zwischenschritte eingibst.

[^^](#) **Wie starte ich ISAC ?**

Doppelklicke das ISAC-Icon auf Deinem Rechner, das Login-Fenster erscheint. Gib Deinen '*username*' (die Nummer Deines Rechners) und Dein '*password*' (wieder die Nummer Deines Rechners) ein.

[^^](#) **Wie wähle ich ein Beispiel und starte eine Rechnung ?**

Du wählst ganz oben im Fenster '*Examples*', bekommst im '*Example browser*' links eine Übersicht über alle Beispiele, die ISAC derzeit rechnen kann, und wählst einen Teil in der Übersicht im Fenster links aus, zum Beispiel '*Algebra Einführung*':

NEW Examples Theories Problems Methods

Example browser
Context Off- >On

- Examples
 - IsacCore
 - Biology
 - Mechanics
 - Statics
 - Etc
 - Algebra Einfuehrung

Wir erkennen, dass wir für 10 cm brauchen, weil jeweils 10 cm abgeschnitten werden müssen.

Unterhalb sind 3 Möglichkeiten dargestellt, die jeweils aus vielen weiteren ausgewählt werden können. Die Kantenlängen sind durch die Farben gekennzeichnet: braun ...

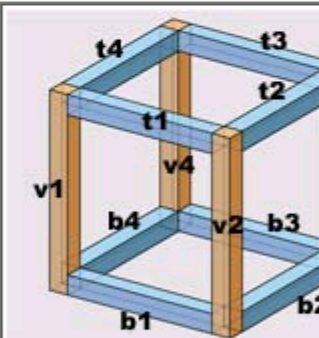


Fig. 2

Die Gesamtlänge, die von 10 cm abgeschnitten werden müssen, ist 10 cm werden:

[rechne Würfel 1](#)

(Bitte schalte im 'Example browser' oben von 'Context on --> off' auf 'Context off --> on', sonst springt der 'Example browser' immer in den [Kontext](#) zur jeweils aktuellen Rechnung.)

Im rechten Teil des [Beispiels](#) findest Du Aufgaben, Erklärungen und Links wie in herkömmlichen eLearning-Materialien. In ISAC gibt es jedoch spezielle Links, die Rechnungen zum Leben erwecken, zum Beispiel der Link ganz unten ['rechne Würfel 1'](#).

Das Demobeispiel zeigt den Teil einer eLearning-Sequenz, in dem diese Aufgabe zu lösen ist: berechne die Länge der Stäbe (Querschnitt 1cm mal 1cm), die zur Herstellung eines Würfels mit Kantenlänge 10cm laut Bild notwendig sind. Der Link '[rechne Würfel 1](#)' startet die zugehörige Rechnung:

The screenshot shows a software interface with a menu bar containing 'NEW', 'Examples', 'Theories', 'Problems', 'Methods', and 'NEXT'. A 'Worksheet' window is open, displaying a file list with the entry 'Problem (Isac.thy, [numerischSymbolische, Berechnung])'. To the right, a problem description is visible, including a diagram of a cube's frame with edges labeled t1, t2, t3, v1, v2, v3, v4, b1, b2, b3, b4. A dimension of 10 is indicated on one edge. Below the diagram is the caption 'Fig. 2'. A blue link labeled 'rechne W' is visible at the bottom right of the page.

Die Rechnung erscheint in einem 'Worksheet', und in diesem steht der Beginn der Rechnung, den wir uns [später](#) genauer anschauen wollen.

[^^](#) **Wie lasse ich ISAC Schritt für Schritt vorrechnen ?**

Gleichzeitig mit dem Worksheet sind auch zwei Schaltflächen 'Next' und 'Auto' erschienen. Drücke 'Next' und ISAC rechnet Dir die Lösung vor, Schritt für Schritt nach Deinem Tempo (in dem Du 'Next' drückst; mit der Schaltfläche 'Auto' kommst Du direkt zum Ergebnis der Rechnung) :

The screenshot shows the ISAC software interface. At the top, there is a menu bar with options: NEW, Examples, Theories, Problems, Methods, and NEXT. Below the menu bar is a window titled "Worksheet" containing a list of mathematical steps for calculating the perimeter of a square frame. The steps are:

- $L = \text{oben} + \text{senkrecht} + \text{unten}$
- $L = t_1 + t_2 + t_3 + t_4 + \text{senkrecht} + \text{unten}$
- $L = k - 2 * q + (k - 2 * q) + (k - 2 * q) + (k - 2 * q) + \text{senkrecht} + \text{unten}$
- $L = 10 - 2 * 1 + (10 - 2 * 1) + (10 - 2 * 1) + (10 - 2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 32 + \text{senkrecht} + \text{unten}$

To the right of the worksheet, there is a diagram of a square frame with labels t_1, t_2, t_3, t_4 for the top edges and v_1, v_2, v_3, v_4 for the vertical edges. A value "10" is shown near the top edge. Below the diagram is the caption "Fig. 2".

Below the diagram, there is a blue button labeled "rechne W".

Wenn Du wissen willst, wie die Rechenschritte zu begründen sind (ganz wesentlich für die mathematische Methode !), überspringe das nächste Kapitel und gehe zum Kapitel [Begründungen](#).

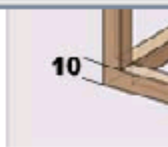
^^ **Wie kann ich selbst Zwischenschritte zur Rechnung eingeben ?**

ISAC kann nicht nur selbst Mathematik, es gibt Dir weitgehend Freiheit, selbst tätig zu werden (und überprüft, ob Du noch richtig liegst). Zum Beispiel kannst Du die nächste Formel in die Rechnung auf dem 'Worksheet' eingeben. Dazu klickst Du 3-mal auf die leere Zeile am Ende der Rechnung (1-mal um sie als aktuelle Zeile zu markieren, und 2-mal um in den Eingabe-Modus zu kommen):

Worksheet

Problem (Isac.thy, [numerischSymbolische, Berechnung])

- $L = \text{oben} + \text{senkrecht} + \text{unten}$
- $L = t1 + t2 + t3 + t4 + \text{senkrecht} + \text{unten}$
- $L = k - 2 * q + (k - 2 * q) + (k - 2 * q) + (k - 2 * q) + \text{senkrecht} + \text{unten}$
- $L = 10 - 2 * 1 + (10 - 2 * 1) + (10 - 2 * 1) + (10 - 2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 32 + \text{senkrecht} + \text{unten}$
- $L = 32 + 4*k + \text{unten}$



nen, dass wir f
en, weil jeweil
nten werden m

sind 3 Möglich
eren ausgewäh
gen sind durch
chnet: braun ..

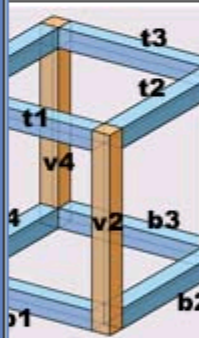


Fig.2

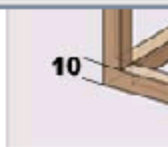
tlänge, die vor
nten werden m

[rechne W](#)

Worksheet

Problem (Isac.thy, [numerischSymbolische, Berechnung])

- $L = \text{oben} + \text{senkrecht} + \text{unten}$
- $L = t1 + t2 + t3 + t4 + \text{senkrecht} + \text{unten}$
- $L = k - 2 * q + (k - 2 * q) + (k - 2 * q) + (k - 2 * q) + \text{senkrecht} + \text{unten}$
- $L = 10 - 2 * 1 + (10 - 2 * 1) + (10 - 2 * 1) + (10 - 2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 32 + \text{senkrecht} + \text{unten}$
- $L = 32 + 4*k + \text{unten}$



nen, dass wir f
en, weil jeweil
nten werden m

sind 3 Möglich
eren ausgewäh
gen sind durch
chnet: braun ..

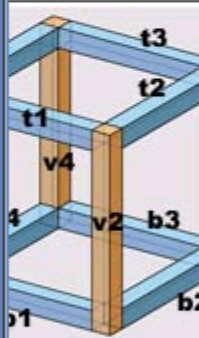


Fig.2

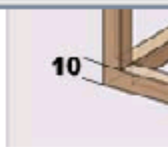
tlänge, die vor
nten werden m

[rechne W](#)

Worksheet

Problem (Isac.thy, [numerischSymbolische, Berechnung])

- $L = \text{oben} + \text{senkrecht} + \text{unten}$
- $L = t1 + t2 + t3 + t4 + \text{senkrecht} + \text{unten}$
- $L = k - 2 * q + (k - 2 * q) + (k - 2 * q) + (k - 2 * q) + \text{senkrecht} + \text{unten}$
- $L = 10 - 2 * 1 + (10 - 2 * 1) + (10 - 2 * 1) + (10 - 2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 32 + \text{senkrecht} + \text{unten}$
- $L = 32 + 4*k + \text{unten}$



nen, dass wir f
en, weil jeweil
nten werden m

sind 3 Möglich
eren ausgewäh
gen sind durch
chnet: braun ..

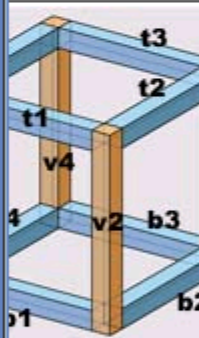
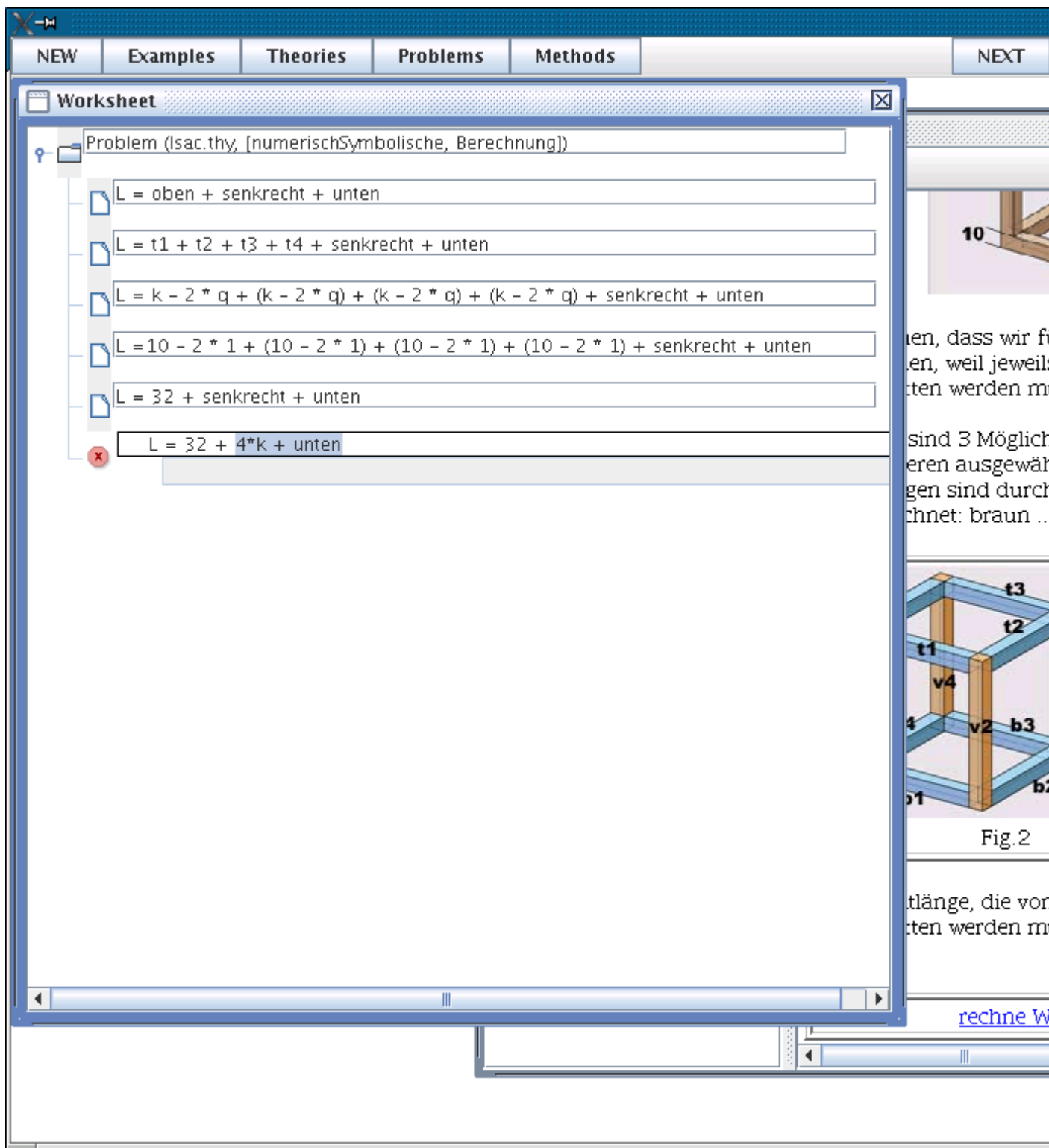


Fig.2

tlänge, die vor
nten werden m

[rechne W](#)



Beende die Eingabe mit der 'Enter'-Taste. Das rote Icon links von der eingegebenen Formel verschwindet, wenn die Formel ohne Schreibfehler ist (fehlende Klammern und Ähnliches) und wenn sie richtig gerechnet ist. ISAC schiebt dann gelegentlich Zwischenschritte oberhalb der eingegebenen Formel ein, wie das nächste [Bild](#) zeigt.

Du hast **völlige Freiheit bei der Eingabe**, solange sie nur richtig ist: Sogar das Ergebnis $L = 104$ kannst Du gleich eingeben (solange ISACs Dialog vom Lehrer nicht anders eingestellt ist ;-)).

Solltest Du Schwierigkeiten haben, eine richtige Formel einzugeben, klicke einfach die vorhergehende Formel und dann 'Next': ISAC sagt Dir eine richtige Möglichkeit !

HINWEISE: Beginne Eingaben in eine Rechnung erst in der zweiten Zeile eines 'Worksheets' (die Zeile, die bereits eingerückt ist), nicht schon in der ersten Zeile.

Tipparbeit kannst Du damit sparen, dass Du die vorhergehende Formel in den Eingabe-Modus nimmst (3-mal klicken), sie mit den 'Ctrl + c'-Tasten in die Zwischenablage kopierst, mit den 'Ctrl + c'-Tasten ins 'Worksheet' unter kopierst und die gewünschten Änderungen vornimmst.

^^ Welche Begründungen gibt es für einen Rechenschritt ?

Mathematik ist jener Teil des menschlichen Denkens, der sich mechanisieren lässt. Und Mathematik ist jene Denkdisziplin, in der jeder Schritt zu begründen ist.

Die direkteste Art, etwas zu begründen ist die (korrekte!) Anwendung einer bestimmten Rechenregel (eines [Theorems](#)). Aber es stellen sich auch Fragen wie 'Warum ist [hier genau diese](#) Rechenregel anzuwenden ?' oder 'Welche ähnlichen Typen von Aufgaben gibt es, und mit welchen [Methoden](#) sind sie zu lösen ?' Die folgenden drei Kapitel zeigen, welche grundlegenden Strukturen ISAC bietet, um mathematische Zusammenhänge zu ergründen.

^^ Die Taktik, die den Schritt erzeugt hat

Die Formeln auf dem 'Worksheet' kommen von einer 'Mathematik-Maschine' (aufbauend auf dem Theoremprover [Isabelle](#)). Die Mathematik-Maschine erzeugt jede Formel mithilfe einer sogenannten Taktik; diese kannst Du für jeden Rechenschritt auf dem 'Worksheet' ansehen, indem Du mit der *linken* Maustaste den Rechenschritt (die Formel) anklickst und dann gleich mit der *rechten* Maustaste das folgende Menü öffnest:

	Übersetzung
Assumptions	Annahmen
Tactic applied	angewandte Taktik
Intermediate steps	Zwischenschritte

Der Menü-Punkt '*Tactic applied*' liefert die jeweils angewandten Taktiken ...

Substitute ["oben = t1 + t2 + t3 + t4"]
Substitute ["t1 = k - 2 * q", "t2 = k - 2 * q", "t3 = k - 2 * q", "t4 = k - 2 * q"]
Substitute ["k = 10", "q = 1"]

... die besagen: 'setze statt k eine 10', 'setze statt q eine 1', etc: Hast Du herausgefunden, warum diese Schritte hier sinnvoll sind zur Lösung der Aufgabe ? Wenn nein: die interaktive Arbeit mit ISAC wird es Dir leicht machen, das herauszufinden.

Der folgende Screenshot zeigt die Taktiken (aber aus technischen Gründen leider nicht das Menü), die aus der 1.Formel die 2., aus der 2.Formel die dritte, und aus der 3.Formel die 4.Formel machen; die Taktiken sind nach rechts gerückt, um sie von den Formeln unterscheiden zu können.

NEW Examples Theories Problems Methods NEXT

Worksheet

Problem (Isac.thy, [numerischSymbolische, Berechnung])

L = oben + senkrecht + unten
Substitute ["oben = t1 + t2 + t3 + t4"]

L = t1 + t2 + t3 + t4 + senkrecht + unten
Substitute ["t1 = k - 2 * q", "t2 = k - 2 * q", "t3 = k - 2 * q", "t4 = k - 2 * q"]

L = k - 2 * q + (k - 2 * q) + (k - 2 * q) + (k - 2 * q) + senkrecht + unten
Substitute ["k = 10", "q = 1"]

L = 10 - 2 * 1 + (10 - 2 * 1) + (10 - 2 * 1) + (10 - 2 * 1) + senkrecht + unten

L = 32 + senkrecht + unten

L = 32 + (v1 + v2 + v3 + v4) + unten

L = 32 + (k + k + k + k) + unten

L = 32 + 4 * k + unten

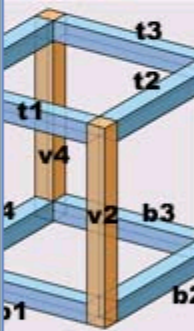


Fig.2

tlänge, die vor
nten werden m

rechner W

^^ Die weiteren Zwischenschritte einer 'Rewrite'-Taktik

Besonders interessant wird es, wenn eine Taktik namens *'Rewrite'* auftaucht; diese Taktiken wenden normalerweise gleich eine ganze Menge von Rechenregeln an (*'Ruleset'*). Diese erzeugen zumeist mehr Zwischenschritte, als dies beim händischen Rechnen geschieht: eine Maschine ist zwar exakt und schnell aber nicht erfinderisch, während ein Mensch zwar erfinderisch aber nicht gerne so exakt ist.

Die Zwischenschritte werden mit dem von oben bekannten [Kontextmenü](#) über die Auswahl von 'Intermediate steps' erzeugt, und zwar die Zwischenschritte zwischen der ausgewählten Formel und der oberhalb von ihr stehenden Formel. Im folgenden Bild wurden die Zwischenschritte für die Formel $L = 32 + \text{senkrecht} + \text{unten}$ erzeugt:

The screenshot shows a software interface with a menu bar containing 'NEW', 'Examples', 'Theories', 'Problems', 'Methods', and 'NEXT'. A 'Worksheet' window is open, displaying a sequence of 11 intermediate steps for a mathematical derivation. The steps are as follows:

- $L = \text{oben} + \text{senkrecht} + \text{unten}$
- Substitute ["oben = t1 + t2 + t3 + t4"]
- $L = t1 + t2 + t3 + t4 + \text{senkrecht} + \text{unten}$
- Substitute ["t1 = k - 2 * q", "t2 = k - 2 * q", "t3 = k - 2 * q", "t4 = k - 2 * q"]
- $L = k - 2 * q + (k - 2 * q) + (k - 2 * q) + (k - 2 * q) + \text{senkrecht} + \text{unten}$
- Substitute ["k = 10", "q = 1"]
- $L = 10 - 2 * 1 + (10 - 2 * 1) + (10 - 2 * 1) + (10 - 2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 10 - 2 * 1 + (10 - 2 * 1) + (10 - 2 * 1) + (10 - 2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 10 + -1 * (2 * 1) + (10 + -1 * (2 * 1)) + (10 + -1 * (2 * 1)) + (10 + -1 * (2 * 1)) + \text{senkrecht} + \text{unten}$
- $L = 10 + -1 * (2 * 1) + (10 + -1 * (2 * 1)) + (10 + -1 * (2 * 1)) + (10 + -2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 10 + -1 * (2 * 1) + (10 + -1 * (2 * 1)) + (10 + -2 * 1) + (10 + -2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 10 + -1 * (2 * 1) + (10 + -2 * 1) + (10 + -2 * 1) + (10 + -2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 10 + -2 * 1 + (10 + -2 * 1) + (10 + -2 * 1) + (10 + -2 * 1) + \text{senkrecht} + \text{unten}$
- $L = 10 + -2 * 1 + (10 + -2 * 1) + (10 + -2) + (10 + -2) + \text{senkrecht} + \text{unten}$
- $L = 10 + -2 * 1 + (10 + -2) + (10 + -2) + (10 + -2) + \text{senkrecht} + \text{unten}$
- $L = 10 + -2 + (10 + -2) + (10 + -2) + (10 + -2) + \text{senkrecht} + \text{unten}$
- $L = 10 + -2 + 10 + -2 + 10 + -2 + 10 + -2 + \text{senkrecht} + \text{unten}$
- $L = 32 + \text{senkrecht} + \text{unten}$

To the right of the worksheet, a diagram of a 3D frame structure is visible, labeled 'Fig. 2'. The diagram shows a rectangular frame with dimensions t1, t2, t3, t4, v1, v2, v3, v4, b1, b2, b3, b4. The number '10' is also present in the diagram area.

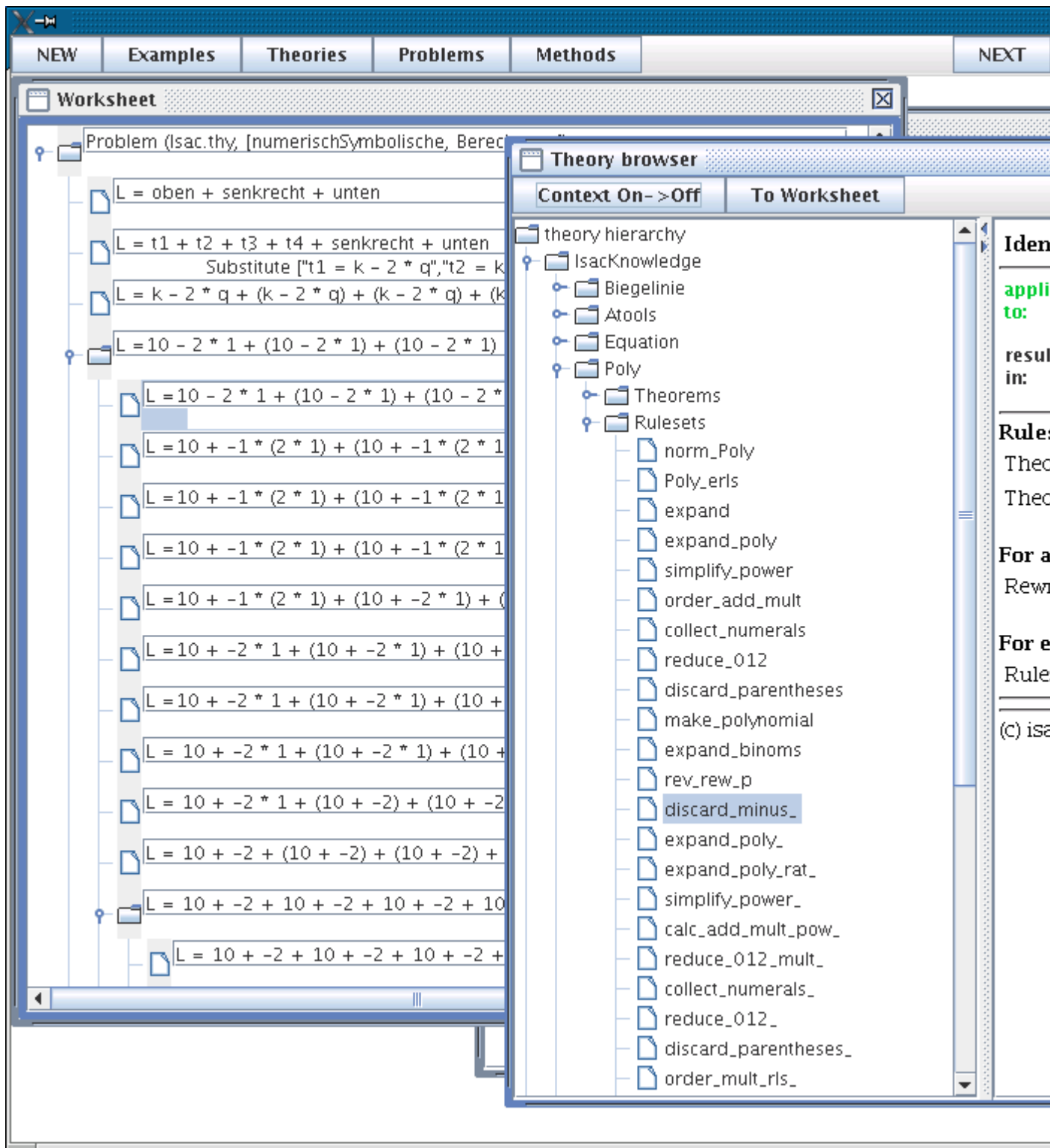
In diesem Beispiel brauchte die Mathematik-Maschine 11 Schritte, wobei es weitere Zwischenschritte zwischen den beiden Zeilen

$$L = 10 + -2 + 10 + -2 + 10 + -2 + 10 + -2 + \text{senkrecht} + \text{unten}$$

$$L = 32 + \text{senkrecht} + \text{unten}$$

gibt. Bitte selbst nachsehen !

Wenn Du nun genau wissen willst, welche Rechenregeln in diesen vielen Schritten angewandt wurden, stelle einen **Kontext** in die zugehörige 'Theorie' her: Klicke die Formel im 'Worksheet' an, von der Du die angewandte Rechenregel wissen willst, und klicke dann die Schaltfläche 'Theories' ganz oben im Hauptfenster an. Das folgende Bild zeigt den **Theorie-Kontext** für die Formel ' $L = 10 - 2 * 1 + (10 - 2 * 1) + (10 - 2 * 1) + (10 - 2 * 1) + \text{senkrecht} + \text{unten}$ ' im 'Theory browser':



Genauerer Hinsehen zeigt: Die Mathematik-Maschine hat eine Regelmeng (‘Ruleset’) namens ‘discard_minus_’ angewandt, die aus zwei Rechenregeln (‘Rules’) namens ‘real_diff_minus’ und ‘sym_real_mult_minus1’ besteht. Klicke auf ‘real_diff_minus’ rechts im ‘Theory browser’ und Du bekommst die Rechenregel (das ‘Theorem’) angezeigt:

<i>real_diff_minus</i>	$?a - ?b = ?a + -1 * ?b$
------------------------	--------------------------

Diese Rechenregel lässt die Minus-Operation verschwinden; es ist eine interessante Frage, warum dies hier geschehen soll (eine schnelle Antwort: weil das ein Vielfaches an Rechenregeln spart: bitte selbst in den Rechnungen nachschauen !). In künftigen Versionen von ISAC wird man auch den Beweis (*Proof*) für dieses Theorem sehen, den der Theoremprover Isabelle mechanisch führt.

^^ **Alles Mathematik-Wissen im Kontext.**

ISACs Mathematik-Wissen umfasst nicht nur Theorien (mit Rechenregeln samt Beweisen, Definitionen, u.ä.) sondern auch [Problem-Typen](#) und [Methoden](#), die die Problem-Typen lösen. Schon jetzt umfasst ISACs Mathematik-Wissen mehr als 2000 Elemente; diese sind nach Gesichtspunkten der Computer-Mathematik geordnet; Du kannst sie alle ansehen, aber in dieser Anordnung sind sie nicht zum Lernen geeignet.

ISAC schafft effiziente Lerngelegenheiten dadurch, dass Du zu einem konkreten Schritt einer bestimmten Aufgabe der angewandten Mathematik einen Kontext in das Mathematik-Wissen herstellen kannst. Das heißt: zu genau diesem Schritt wird die angewandte Rechenregel gezeigt, zu dieser Aufgabe wird der passende Problem-Typ angezeigt, und zu diesem Problem-Typ die Methode(n) zur Lösung der Aufgabe.

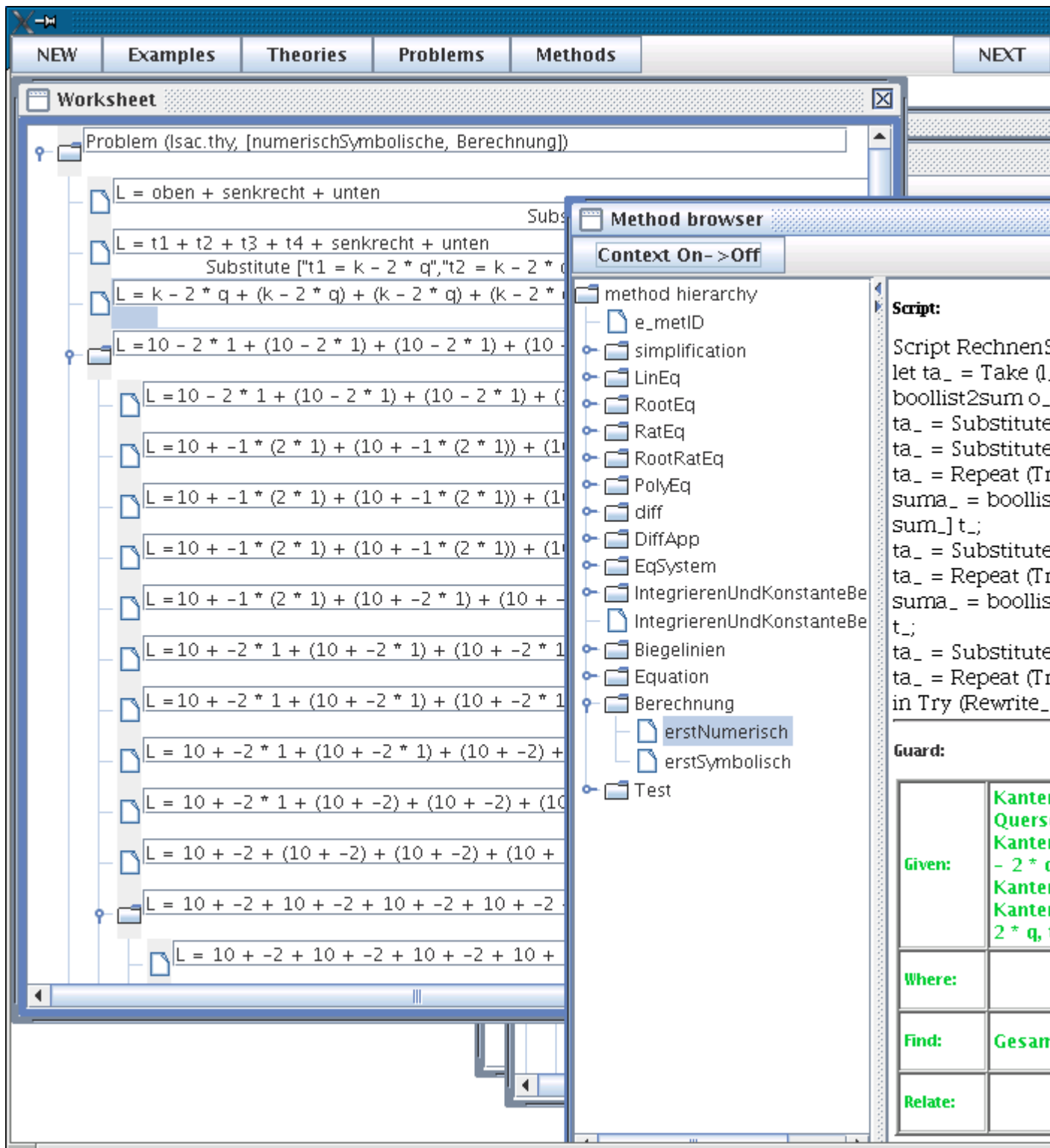
Damit kannst Du Fragen wie 'Warum ist hier genau diese Rechenregel anzuwenden ?' oder 'Welche ähnlichen Typen von Aufgaben gibt es, und mit welchen Methoden sind sie zu lösen ?' nachgehen.

Die folgenden zwei Kapitel geben Beispiele dafür.

^^ **Woher weiß ISAC seine Taktiken ?**

Nach den Informationen bisher scheint ISAC eine universelle Problem-Löse-Maschine zu sein. Die Mathematiker haben aber längst bewiesen, dass es eine solche nicht geben kann !

Woher also bekommt ISAC seine Problemlöse-Kapazität ? Die Antwort auf die Frage ergibt sich wieder daraus, dass ISAC alle Elemente seines Wissens für Dich zugänglich macht: Die Schaltfläche '*Methods*' öffnet den '*Method browser*':



Der 'Method browser' zeigt nun jene **Methode**, aus der ISAC die [Taktiken](#) für die aktuelle Rechnung genommen hat. In unserem Beispiel sind dies die 'Substitute'-Taktiken, gefolgt von weiteren Taktiken (Achte darauf, dass im 'Worksheet' noch die Würfel-Aufgabe steht, dann hat sich der 'Method browser' im Kontext zu dieser Aufgabe geöffnet; er zeigt die Methode und ihr 'Script' (das Programm). In künftigen Versionen von ISAC wird jene Taktik im 'Script' hervorgehoben, die die im 'Worksheet' angeklickten Schritt erzeugt hat.

Jede Methode enthält neben dem *'Script'* auch einen *'Guard'*, der die richtige Anwendung der Methode überwacht.

^^ **Woher weiß ISAC, aus welcher Methode die Taktiken zu nehmen sind ?**

Nun scheint nur mehr die Frage offen, woher ISAC weiß, welche die passende Methode für die gewählte Aufgabe ist.

Ganz einfach: Hinter jedem Aufgaben-Text im *'Example browser'* steht unsichtbar ein Modell mit den notwendigen Formeln und Verweise in das 3-dimensionale Mathematik-Wissen: ein Verweis auf die zugehörige Theorie, einer auf den Problem-Typ und einer auf die Methode zur Lösung der Aufgabe.

Das [Kontext-Menü](#) auf der ersten Zeile eines Worksheets (und auf Sub-Problemen) hat einen weiteren Punkt zur Auswahl:

	Übersetzung
Assumptions	Annahmen
Tactic applied	angewandte Taktik
Intermediate steps	Zwischenschritte
Calthead	Beschreibung einer Aufgabe

Mit *'Calthead'* öffnet man ein Fenster (namens *'CalcHeadPanel'*) im *'Worksheet'*, mit dem Modell (*'Model'*) und mit Verweisen (*'Specification'*) auf Theorie, Problem und Methode. Diese Art, eine Aufgabe festzulegen, ist die allgemeinste, kürzeste und präziseste aller Möglichkeiten.

Das Bild unterhalb zeigt den *'Problem browser'* im Kontext. Diesen erhält man über die Schaltfläche *'Problems'* oben im Hauptfenster.

^^ Kann ich selbst neue Rechnungen eingeben ?

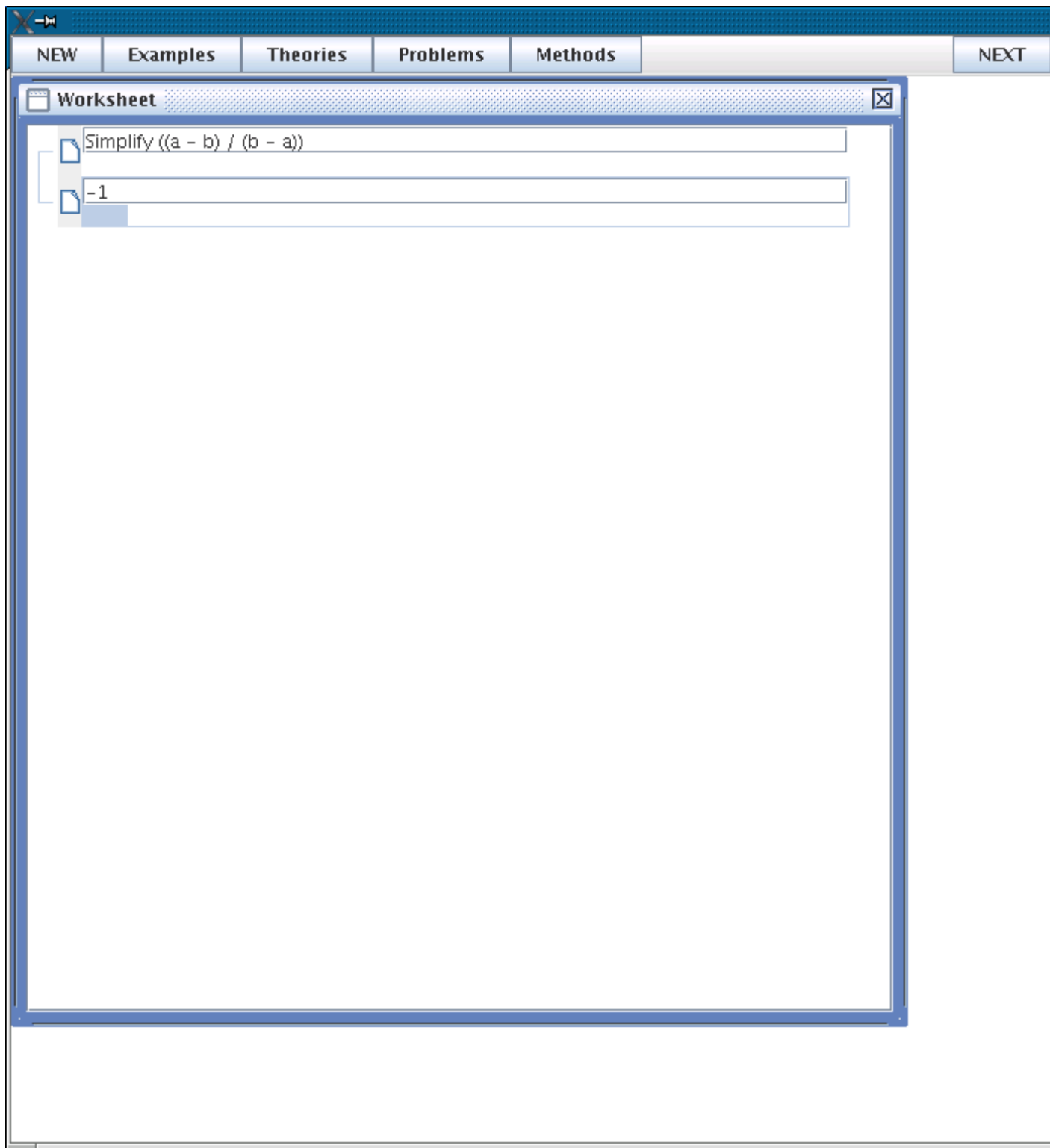
Ja.

Das oben gezeigte Beispiel macht klar, dass dazu zuerst eine Theorie, ein Problem und eine Methode zu wählen und ein *'Model'* einzugeben sind; wir zeigen das hier nicht (es sollte sich aber durch Probieren herausfinden lassen).

Hier zeigen wir etwas Einfacheres: wie sich ISAC als Algebra System benutzen lässt -- mit den bekannten Funktionsaufrufen:

Eingabe	Bedeutung	Beispiel
<code>solve (e_::bool, v_)</code>	löse die Gleichung $e_$ nach der Variablen $v_$	<code>solve (x+1=2, x)</code>
<code>solveSystem es_ vs_</code>	löse die Liste von Gleichungen $es_$ nach der Liste von Variablen $vs_$	<code>solveSystem [a+</code>
<code>Diff (f_, v_)</code>	differenziere den Term $f_$ nach der Variablen $v_$	<code>Diff (x^2+x+1, x)</code>
<code>Differentiate (f_, v_)</code>	differenziere die Funktionsgleichung $f_$ nach der Variablen $v_$	<code>Differentiate (P =</code>
<code>Integrate (f_, v_)</code>	integriere die Funktionsgleichung $f_$ nach der Variablen $v_$	<code>Integrate (x^2+x</code>
<code>Simplify t_</code>	simplifiziere den Term $t_$	<code>Simplify ((a - b)/</code>

Die Schaltfläche *'New'* öffnet ein leeres *'Worksheet'*. In dieser ersten Zeile lässt sich ein Funktionsaufruf eingeben. Das letzte Bild dieser Kurzeinführung zeigt die unterste Funktion der Tabelle oben nach Betätigen der Enter-Taste und der Schaltfläche *'Auto'*.



Viel Spaß beim Experimentieren mit ISAC !

^^ Zusammenfassung

ISACs neuartige Eigenschaften lassen sich am einfachsten fassen, wenn man sie mit denen eines Algebra-Systems (zum Beispiel Math-CAD) vergleicht:

- ISAC umfasst die Funktionen eines Algebra Systems (*'New'*, [Funktion](#) eingeben, *'Auto'* drücken); die Leistungsfähigkeit eines kommerziellen Algebra Systems hat ISAC noch nicht erreicht.
 - ISAC hat eine allgemeinere Art der Beschreibung ([Spezifikation](#)) von Aufgaben. Damit kann ein Mathematik-Autor beliebige Aufgaben aus angewandter Mathematik so beschreiben, dass ISAC sie mechanisch lösen kann.
 - ISAC zeigt nicht nur das Ergebnis, sondern auch die [Zwischenschritte](#). Die Zwischenschritte entsprechen weitgehend jenen von händisch gerechneten Rechnungen.
 - Zu jedem Rechenschritt kann man das gesamte [Mathematik-Wissen](#) einsehen, das ISACs Mathematik-Maschine verwendet hat (ISAC ist ein 'transparentes' System). Da diese Maschine nach den Regeln der formalen Logic arbeitet, lassen sich auch komplexe Begründungen herausfinden.
 - ISAC unterstützt die freie Gestaltung von Rechenschritten durch die Möglichkeit, (fast) jederzeit die Formel zu diesem Schritt einzugeben. Die Korrektheit einer Formel überprüft ISAC unfehlbar in algebraischen Bereichen mit eindeutiger Normalform (wie sie überwiegend in der angewandten Mathematik auftreten).
-