Using Symbolic Problems in Teaching Chemistry

Attilio Immirzi[†]

Dipartimento di Chimica Università di Salerno, Via S. Allende, Baronissi (SA), I-84081 Salerno, Italy

Received April 3, 1997

A computer program is illustrated which translates and randomizes "symbolic" chemistry problems so producing thousands of tasks from a few ones, each with the correct solutions. The procedure could be the basis for implementing a "Fully Automatic Examination System" based on individual tasks *always different from each other*. The program handles symbolically not only numerical figures but also the substances themselves and retrieves physical properties from special files so as to avoid unrealistic or arbitrary exercises. By using the T_EX text-processor one gets tidy forms ready to be used in examinations and very simple to verify.

INTRODUCTION

The rising number of university students and the decreasing level of culture in the high schools always make teaching a more frustrating activity. I become even more convinced that is it indispensable to compel the students in resolving more and more practical problems, until they grasp the concrete meaning of physical laws and learn to manage correctly physical quantities, units, conversion factors, etc. All people agree, so I believe, but, in practice, teachers with hundreds of students cannot spend their lives preparing and correcting exercises. Furthermore, at least in my country, students have an incurable tendency to "collaborate among themselves" with the final result that examination papers are not only generally wrong but also quite similar to each other! To overcome this problem a solution exists: assigning to each student his/her own problem; the correction work would become however really prohibitive!

In brooding over this problem for a long time, I have finally found a practical solution. As a matter of fact most chemistry problems reduce to a number of "basic" cases; one can create thousands of different tasks from a few emblematic cases. Thus I have developed this idea: to use a computer to multiply a "symbolic" problem to thousands of problems, really similar to each other but different for numerical figures and also in the choice of substances. The same computer must supply, of course, not only the translated (and randomized) problems but also the solutions! The teacher can quickly perform the correction, merely comparing the proposed solution with the one calculated by the computer and stored in a reserved file.

By creating a set of a few hundred symbolic problems, well classified by argument and difficulty level, one can easily implement an automatic system, which selects at random a number of symbolic problems, translates, and randomizes them giving to symbols real figures, and finally prepare, student by student, a standardized ready-to-use examination form.

[†] E-mail: IMMIRZI@MVXCHE.CSIED.UNISA.IT Fax: 0039-89-965.296.

DESCRIPTION OF THE PROGRAM

The program STT (Symbolic Task Translator) has been implemented as the "heart" of a future automated system described above. STT is written in Fortran language (NDP Fortran compiler) and runs on any DOS-PC with a 386 (or higher) mainboard. Present size is about 2000 statements.

An essential complement of the program consists of a nomenclature file containing scientific or common names of the substances mentioned and a number of files which contain physical and chemical properties of the substances themselves. The latter properties (data were taken from CRC Handbook of Physics and Chemistry[1]) are retrieved by STT and used by means of symbolic names, as shown below. Program output is not a conventional printout, but an ASCII file structured according to T_EX rules[2] so that tidy examination forms can be directly produced. The principal features are well illustrated below by a few simple examples. The symbolic problem shown in Chart 1 might produce for instance this task *written exactly as shown* on the student form (solutions are also given; they are of course omitted in real cases!):

> Treatment of magnesium bicarbonate with aqueous hydrogen chloride affords magnesium chloride according to the reaction (to be balanced!)
>
> $$Mg(HCO_3)_2 + HCl \rightarrow MgCl_2 + CO_2 + H_2O$$
>
> The mass of $Mg(HCO_3)_2$ treated is 10.47 g and aqueous HCl is 24% mass. Balance the reaction and establish (1) the mass of aqueous acid to be used (g); (2) the volume of CO_2 obtained (liters measured at 31 °C and 291 mmHg pressure); (3) the mass of $MgCl_2$ resulting (g). Solution no. 1: 21.74 g of HCl at 24%. Solution no. 2: 9.327 L of CO_2. Solution no. 3: 6.81 g of $MgCl_2$ salt.

The meaning of the input data is as follows (characters after '!' are comments):

(1) PROB record indicates the beginning of a problem. It is followed by the symbolic text written according to T_EX

S0095-2338(97)00022-X CCC: $15.00
Published on Web 07/18/1998

Chart 1

```
PROB
Treatment of #UNa with aqueous #UNb affords #UNc according to the reaction
(to be balanced!)
$$  #MLa \pl #MLb \gv #MLc \pl #MLd \pl #MLe   $$
The mass of #MLa \ treated is #gra g and #MLb \ is used as aqueous solution
(#pcb\% mass).  Balance the reaction and establish
(1) the mass of aqueous acid to be used (g); (2) the volume of \cx CO_2 \
obtained (liters measured at #tec \degc\ and #phg mmHg pressure);
(3) the mass of #MLc \ resulting (g).
\gsl 1 {#grb g of #MLb \ at #pcb\%}
\gsl 2 {#vlo L of \cx CO_2 }
\gsl 3 {#grc g of #MLc \ salt}
SOLN
SEL MLa Ca(HCO_3)_2  Mg(HCO_3)_2 KHCO_3  Sr(HCO_3)_2 Ba(HCO_3)_2 NaHCO_3
COU MLb H_3PO_4      HCl          H_2SO_4 H_3PO_4     HBr          H_3PO_4
COU MLc Ca_3(PO_4)_2 MgCl_2       KHSO_4  SrHPO_4     BaBr_2       Na_3PO_4
SEL MLd CO_2
SEL MLe H_2O
SEL gra 5.40 12.40  0.01                !  mass of bicarbonate (g)
SEL pcb 20  40  1                       !  % of MLb acid
SEL tec 15  32  1                       !  temperature (C)
SEL phg 200 500 1                       !  pressure (mm Hg)
BRC MLa + MLb = MLc + MLd + MLe         !  balance chemical reaction
nma=gra/Mma                             !  g-moles of bicarbonate
nmb=nma*(MCb/MCa)                       !  g-moles of acid
nmc=nma*(MCc/MCa)                       !  g-moles of salt formed
nmd=nma*(MCd/MCa)                       !  g-moles of CO_2 formed
nme=nma*(MCe/MCa)                       !  g-moles of H_2O formed
grb=nmb*Mmb/(0.01*pcb) (F8.2)           !  mass of acid in solution (g)
grc=nmc*Mmc            (F8.2)           !  mass of salt formed (g)
vlo=nmd*Rgl*(tec+Tze)/(phg/760) (F9.3)  !  volume of CO_2 evolved (litres)
END
```

rules. In the text a number of strings of format #xyz appear which will be replaced by real quantities at run time. Strings like #UNx will be replaced by names of substances; strings like #MLx will be replaced by chemical formulas. Other strings will be replaced by numerical quantities, either input data (more or less randomly generated, see later) or output data (calculated by the program). A number of these symbols (beginning with a capital letter) are either general physical quantities or specific ones for the selected substances, either calculated or retrieved from special files (see point 3 and further examples). There are in the text also references to useful macros like \cx (writing chemical formula), \degc (for °C), and so on. \gsl is a macro for writing the results.

(2) String SOLN indicates the end-of-text and the beginning of solution. It is followed by "commands" of various kinds:

(3) SEL MLx command (x = a, b, c, ...p) selects at random a substance among the following (continuation cards are allowed) and attaches to it the label MLx. Once the substance is defined the program looks for the common name of the substance recorded in a special file, analyzes the formula, and establishes atomic composition and molecular mass. In many cases (see next examples) specific properties for the selected substances (e.g., vapor pressure data, thermodynamic quantities, dissociation constants, etc.) are retrieved from special files and become usable by conventional labels. So Mmx is the molecular mass for MLx molecule, Wax and Wbx are, for instance, the *a*,*b* van der Waals constants of a real gas, Dcx is the dissociation constant (if applicable!), etc. Universal constants, numerical constants, conversion factors, atomic masses, etc. are also read from a file and become accessible with conventional symbols: Rg1 is, for example, the gas *R* constant in L atm mol^{-1} K^{-1}, Avn is the Avogadro's number, etc.

(4) COU MLx command also selects a substance among the given list, not taking it at random however, but taking the item in the same position of the one selected above. This allows, in the given example, selecting together with say the bicarbonate $Mg(HCO_3)_2$, the acid HCl, and the salt $MgCl_2$.

(5) SEL xyz command (xyz is any three-character string) assigns at random a value to the "variable" labeled xyz with lower and upper limits indicated in the two next fields and increments given by the third field. The limits are constant values in the example; they can also be symbols or expressions (see below, point 7).

(6) BRC command (BRC is followed by a chemical equation involving MLx symbols) means balance chemical

Chart 2

```
PROB
The saturated solution of the sparingly soluble #UNa at 25 \degc\ has osmotic
pressure of #osm mmHg.  Calculate the solubility product of #UNa, neglecting,
for simplicity, possible hydrolysis effects.
\gsl {#Spa}
SOLN
OPN proper.002
SEL MLa  CaF_2  CaSO_4  BaSO_4  Ag_2CrO_4  Ag_3PO_4  PbSO_4  PbI_2  Cu_2S  &
ZnS  CdS    FeS     CuSCN   PbCO_3     NiS       MnS     CuBr   CuI
pow=1.0/(Npa+Nma)                !  Npa = no.  of cations; Nma = no.  of anions
den=Npa^Npa+Nma^Nma
sol=(Spa/den)^pow                !  molar solubility
cef=sol*(Npa+Nma)                !  effective total molarity
osm=cef*Rgl*(25+Tze)*760.0 (F9.2) !  osmotic pressure (mm Hg)
END
```

reaction. Reaction coefficients (integers without common divisors) are determined and set-up as MCa, MCb, etc.

(7) The following are executable commands: all are of the type xyz=expression the latter being a combination of variables, numerical constants, operators, and functions obeying rules similar to Fortran's rules. All variables referenced in expressions are checked at run time before statement execution to avoid undefinite values. When calculated quantities must appear in the text, a formatting field (e.g., F9.3) is also given.

(8) The executable commands are performed in sequence. The natural sequence can, however, be altered by a JMP command (there are not in the above example, see next ones) which allows unconditioned or conditioned jumps.

After the execution of the last command (command stream terminates with an END command) the preceding T_EX text is examined, and all symbols beginning with # are replaced by the corresponding quantities. The new text is outputted and is ready to be processed by T_EX.

The next very simple task illustrates how specific properties (e.g., solubility products) are retrieved from special files. Furthermore the example shows a very useful trick, viz., to start from "solution" and obtain "initial" data. The symbolic problem shown in Chart 2 could produce, for instance, the following problem:

> The saturated solution of the sparingly soluble silver chromate at 25 °C has osmotic pressure of 3.38 mmHg. Calculate the solubility product of silver chromate, neglecting, for simplicity, possible hydrolysis effects. Solution no. 1: 1.11×10^{-12}.

In this task the command OPN proper.002 appears which means "search in proper.002 file possible quantities belonging to selected substances". As shown problem selects a substance MLa (among numerous sparingly soluble salts) and makes use of some variables (Npa, Nma, Spa) whose values are just retrieved from the file above (Npa = number of cations in MLa, Nma = number of anions, Spa = solubility product).

The next case illustrates how STT may truly become a deadly weapon in the hands of perverse teachers! The problem is really straightforward if "initial" and "final" data are interchanged, while it is very tedious for the student who must resolve the task by trial-and-error. The symbolic problem shown in Chart 3 could produce, for instance, the following problem:

> Calculate for a mixture of carbon tetrachloride (standard boiling temperature: 76.7 °C) and 1,1,2,2-tetrachloro-1,2-difluoroethane (standard boiling temperature 92.0 °C), consisting in 47 mass percent of carbon tetrachloride, the temperature at which the overall vapor pressure for the mixture is 810.42 mmHg. CCl_4 has a vapor pressure of 400 mmHg at 57.8 °C, and CCl_2-$FCCl_2F$ has the same vapor pressure at 73.1 °C. Use the Clausius−Clapeyron equation (with constant ΔH_{ev}) for each component and assume that the mixture obeys Raoult's law for each composition. Solution no. 1: 85.0 °C.

The solution is indeed simple: proper.005 file contains, for many molecular substances, the six temperatures (Vax, Vbx, ...Vfx) for which vapor pressure is, respectively 1, 10, 40, 100, 400, 760 mmHg. STT chooses two molecular substances MLa and MLb and a random temperature tmp between the two boiling points and then evaluates vapor pressures for the two pure substances tva, tvb by means of the Clausius−Clapeyron equation and finally the vapor pressure ovp for the mixture by means of Raoult's law. This last value is formally an input datum, while the above tmp temperature is an output datum.

Let us now consider the last case shown in Chart 4 a problem based on van der Waals equation[3] which requires iterations. SST could produce, for instance, this text:

> A gas reservoir of 2.7 m^3 volume is filled with gas methane. The temperature can rise up to 59 °C and the pressure up to 180 atm. Calculate the maximum load of gas in kilograms compatible with the above upper limits for temperature and pressure, assuming that methane obeys the van der Waals equation. Constants are $a = 2.253$ atm L^2 mol^{-2}, $b = 0.04278$ $Lmol^{-1}$. Solution no. 1: 286.1 kg.

As the van der Waals equation $(P + n^2a/V^2)(V - nb) = nRT$ can be put in the form $P = f(n, V, T)$ but not in the form $n = f(P,V,T)$ the analytical solution of n vs P, V, T *is* not allowed, and some iterative procedure is necessary (e.g., trying n values by Newton method[5]). The command JMP xyz nl n2 n3 just allows iterations as it alters the natural command sequencing: control goes to the current statement

Chart 3

```
PROB
Calculate for a mixture of #UNa (standard boiling temperature:  #Vfa \degc )
and #UNb (standard boiling temperature #Vfb \degc), consisting in #pca mass
percent of #UNa, the temperature at which overall vapor pressure for the
mixture is #ovp mm  Hg.  #MLa \ has a vapor pressure of 400 mm Hg at #Vea
\degc\ and #MLb \ has the same vapor pressure at #Veb \degc.
Use Clausius-Clapeyron equation (with constant $\Delta H_{ev}$) for each
component and assume that the mixture obeys Raoult's law for each composition.
\usl{#tmp} {\degc}
SOLN
OPN proper.005
SEL MLa CCl_4         CH_2Cl_2  CH_3I  TiCl_4  SbCl_3  Cd
COU MLb CCl_2FCCl_2F  CH_2Br_2  CH_3F  VOCl_3  SbBr_3  Zn
SEL tmp Vfa+1 Vfb-1 1.0    !  N.B. MLa must be more volatile than MLb !!
SEL pca 20 80 1            !  mass percent of MLa
pcb=100-pca                !  mass percent of MLb
COM ln(p2/p1)=(Dh/R)*(1/T1-1/T2)
dha=LOG(760/400)/(1.0/(Vea+Tze)-1.0/(Vfa+Tze)) !  Dh/R for MLa
dhb=LOG(760/400)/(1.0/(Veb+Tze)-1.0/(Vfb+Tze)) !  Dh/R for MLb
tva=400*EXP(dha*(1.0/(Vea+Tze)-1.0/(tmp+Tze))) !  v.p.  at tmp for MLa
tvb=400*EXP(dhb*(1.0/(Veb+Tze)-1.0/(tmp+Tze))) !  v.p.  at tmp for MLb
nma=pca/Mma !  number og g-moles of MLa
nmb=pcb/Mmb !  number og g-moles of MLb
ovp=tva*nma/(nma+nmb)+tvb*nmb/(nma+nmb) (f12.2) !  overall vapor pressure
END
```

Chart 4

```
PROB
A gas reservoir of #vmx \mc\ volume is filled with gas #UNa.  The temperature can
rise up to #tmx \degc\ and the pressure up to #pmx atm.  Calculate the maximum
load of gas in kilograms compatible with the above upper limits for temperature and
pressure, assuming that #UNa obeys the van der Waals equation.  Constants
are  $a$ = #Waa $\rm atm\,L^2\,mol^{-2}$, $b$ = #Wba $\rm L\,mol^{-1}$.
\gsl 1 {#kga kg}
SOLN
OPN proper.001        !  access to file containg Van der Waals and critical data
SEL MLa  Ar  N_2  O_2  CO_2  CH_4  C_2H_6
SEL vmx  2.0 6.0 0.1  !  select reservoir volume (m^3)
SEL tmx  40 70 1      !  select max temperature (C)
SEL pmx  100 400 10   !  select max pressure (atm)
vlt=1000.0*vmx        !  volume in litres
nim=pmx*vlt/(Rgl*(tmx+Tze))  !  approximate no.  of g-moles (Clapeyron equation)
ptr=(nim*Rgl*(tmx+Tze))/(vlt-nim*Wba)-nim^2*Waa/vlt^2   !  trial P
der=Rgl*(tmx+Tze)*vlt/(vlt-nim*Wba)^2-2.0*nim*Waa/vlt^2 !  dP/dn
nex=(pmx-ptr)/der         !  increment for nim approaching ptr to pmx
chk=ABS(nex/nim)-0.001    !  fractional increment
nim=nim+nex               !  increasing nim
JMP chk 1 1 -5            !  if chk<0 the trial is satisfactory; else repeat
kga=0.001*nim*Mma (f12.1) !  gas load in kg
END
```

+nl or +n2 or +n3 when the xyz expression is negative, null, or positive.

Other useful commands are allowed or are under development: the command STF MLa MLb ..., for instance, set-up, as a function of temperature, the thermodynamic functions enthalpy, entropy, and free energy[3] for all selected molecules retrieving in a special file formation enthalpies, absolute entropies, and thermal capacities. Another useful feature is the calculation of equilibrium composition for reacting gas mixtures performed with the command SEQ.

CONCLUSIVE REMARKS

The illustrated program, though fitted with a limited number of symbolic problems and incomplete "properties files", is representative of a promising tool for the realization of an "Automated Examination System".

A remarkable advantage is that the system should be fully expandable by each user, as anyone can expand and customize the problem file creating its own problems. Provided that a standard protocol is obeyed problems could be interchanged among chemistry teachers; this should allow creating a very large set of symbolic problems potentially multipliable in the millions.

In order to make an efficient system it is anyway essential to assemble a wide set of symbolic tasks, particularly if exercises spanning many arguments and a wide range of difficulty are desired. Much time and teamwork is desirable also because problems must be carefully tested one-by-one. The interest of software producers or educational institutions is desirable.

REFERENCES AND NOTES

(1) *CRC Handbook of Chemistry and Physics*, 65th ed.; CRC Press: 1985.
(2) Knuth, D. E. *The Textbook*; Addison Wesley: Reading MA, 1988.
(3) Atkins, P. A. *Physical Chemistry*; Oxford University Press: 1982.
(4) Abramovitz, M.; Stegun, I. A. *Handbook of Mathematical Functions*; Dover Publications: 1972.

CI970022D