HHS Public Access

Author manuscript

*J Chem Inf Model*. Author manuscript; available in PMC 2021 October 07.

Published in final edited form as:

*J Chem Inf Model*. 2020 July 27; 60(7): 3336–3341. doi:10.1021/acs.jcim.0c00448.

# Adapting CHMTRN (CHeMistry TRaNslator) for a New Use

**Philip N. Judson**,
Heather Lea, Norwood, Harrogate HG3 1TE, England

**Wolf-Dietrich Ihlenfeldt**,
Xemistry GmbH, D-61479 Glashütten, Germany

**Hitesh Patel**,
Computer-Aided Drug Design Group, Chemical Biology Laboratory, Center for Cancer Research, National Cancer Institute, National Institutes of Health, Frederick, Maryland 21702, United States

**Victorien Delannée**,
Computer-Aided Drug Design Group, Chemical Biology Laboratory, Center for Cancer Research, National Cancer Institute, National Institutes of Health, Frederick, Maryland 21702, United States

**Nadya Tarasova**,
Synthetic Biologics and Drug Discovery Group, Laboratory of Cancer Immunometabolism, Center for Cancer Research, National Cancer Institute, National Institutes of Health, Frederick, Maryland 21702, United States

**Marc C. Nicklaus**
Computer-Aided Drug Design Group, Chemical Biology Laboratory, Center for Cancer Research, National Cancer Institute, National Institutes of Health, Frederick, Maryland 21702, United States

## Abstract

We have adopted and extended the CHMTRN language and used it for the knowledge base of a computer program to generate a large database of synthetically accessible, drug-like chemical structures, the Synthetically Accessible Virtual Inventory (SAVI) Database. CHMTRN is a powerful language originally developed in the LHASA (Logic and Heuristics Applied to Synthetic Analysis) project at Harvard University and used together with the chemical pattern description language, PATRAN, to describe chemical retro-reactions. The languages have proven to be useful beyond the design of retrosynthetic routes and have the potential for much wider use in chemistry; this paper describes CHMTRN and PATRAN as now reimplemented for the forward-synthetic SAVI project but able to describe both forward and retro-reactions.

## Graphical Abstract

**Corresponding Author Philip N. Judson** − *Heather Lea, Norwood, Harrogate HG3 1TE, England;* philip@judsonpn.uk.

Notes

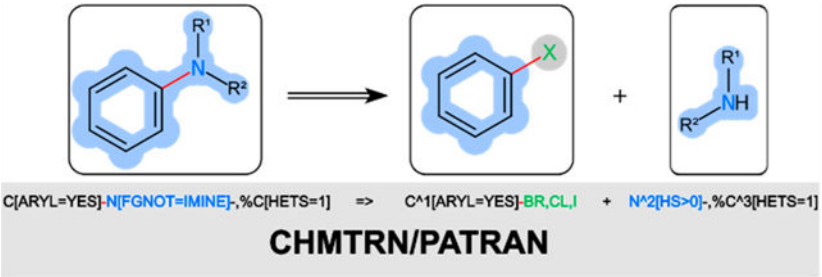
The authors declare no competing financial interest.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c00448.

Tables of terms used in CHMTRN and PATRAN and further details about them (PDF)




## INTRODUCTION

Virtual screening has become a common step in early stage drug discovery. However, screenable databases of compounds are still dwarfed by the size of the universe of drug-like molecules.[1] We and others[2–5] aim to generate databases of billions of compounds that can be synthesized in one step from readily available starting materials, which constitutes a significant challenge. For our project, the Synthetically Accessible Virtual Inventory (SAVI) Database, it was necessary to create and maintain a generic chemical reactions knowledge base.[6–9] Use of a text-based language was preferred to a graphical tool both because of the potential savings in programming effort and because it would allow anyone to work on, and read, knowledge base content using only a text editor. For a while in the 1970s the use of natural language like programming languages was favored. This idea went out of fashion but, with the rapid development of natural language processing more generally, it is perhaps time to look at it again.

CHMTRN, originally developed by a team at Harvard University for the LHASA (Logic and Heuristics Applied to Synthetic Analysis) project in the 1970s,[10,11] has been adopted and adapted for the SAVI project. The purpose of this paper is to describe those parts of the revised language that we are using. A more extended version of this Application Note is available at 10.26434/chemrxiv.11439984.v1.

## BACKGROUND

When CHMTRN was created by the team at Harvard University, it was extremely forward looking. Its coverage and flexibility still do not appear to be matched by any other textual language for coding reaction and retro-reaction descriptions.

In the LHASA project, a software interpreter packed the information contained in the knowledge base into a set of computer words for fast processing at run time. We have developed a new, similar software interpreter, which we describe in a separate paper about the SAVI project,[6] using the CACTVS toolkit.[12–15] This paper is about the language itself. We have adopted those terms used in the original CHMTRN that are relevant to our project and added new ones, including modifications to make the language more convenient for use with applications carrying out reaction prediction as well as those carrying out retrosynthetic analysis. The definitions of many terms in the language as implemented for SAVI are closely similar to those used in the original CHMTRN, since the terms are everyday ones in chemistry, such as “acid halide”, “double bond”, “leaving group”. Nevertheless, in the

Supporting Information to this paper we have defined how all of them are used in SAVI to avoid potential uncertainty.

## CHMTRN AND PATRAN

Each description of a retrosynthetic reaction in CHMTRN is termed a “transform”. A central purpose of CHMTRN is to allow reasoning about sets (e.g., a conditional statement based on whether an atom belongs to a set with a particular property). It provides most of the control and conditional functionality expected of a modern programming language, such as “if ⋯ then … else”, and “for each” statements. The writer can instruct the application reading the knowledge base to adjust its confidence in the likely success of a reaction, depending on chemical structure and functionality in a query, and to issue messages to the end user.

Transforms in the CHMTRN developed for the LHASA project could contain a linear string description of the keying features of the reaction product (i.e., the target in a retro-reaction) written in PATRAN (*vide infra*).[16] We have extended PATRAN to describe also the precursors to the retro-reaction, thus allowing transforms to be run in either direction. This was true for LHASA transforms that included “2D” patterns (described briefly in the Supporting Information), but the 2D patterns were more difficult to design and enter and provided less support for specifying atom and bond properties than the so-called “1D” PATRAN strings.

LHASA transforms were written retrosynthetically. Transforms for the SAVI project could have been written in the synthetic direction with only the minor change of introducing a forward reaction arrow into the PATRAN language and small changes to how multiple reaction components are handled but we have written them retrosynthetically for backward compatibility with the LHASA program.

### Transform, Atom, and Bond Identifiers.

The first line of each transform description is a header defining a unique, integer identifier for the transform.

e.g., TRANSFORM 1234

Atoms and bonds that are included in the keying substructure(s) are uniquely identified within a transform by the terms ATOM*n and BOND*n, where n is an integer between 1 and 32. Note that while an atom or bond is unique within a transform, it may be present in both the target (product) and a precursor (reactant). For example the oxygen atoms in the target and the first precursor in the following retro-reaction are the same atom:

$$R{-}O{-}R' => R{-}OH + R' - Br$$

### Hydrogen Atoms.

In CHMTRN, hydrogen atoms are not treated as atoms in their own right, except where this is necessary for particular reasons, but as properties of the atoms to which they are attached. The implications of this are discussed in the Supporting Information.

## PATTERNS

In early implementations of the CHMTRN language, transforms were keyed by functional groups in a target structure (product). Later PATRAN, a flexible, string representation for keying substructures to be read by a PAttern TRANslator module, was developed and incorporated into transform descriptions under the header, 1D*PATTERN. Patterns contained information only about the substructural features associated with the product of a reaction since LHASA operated retrosynthetically. To support operating also in the forward-synthetic direction, a new representation was developed and incorporated into transforms under the header, 2D*PATTERN. As explained below, for our current project we have instead extended the linear 1D*PATTERNs to describe the precursors as well as the products of reactions.

Pattern statements in a transform are preceded by the name of the pattern type and a carriage return. They are followed, after the last pattern, by END*PATTERN or END*PATTERNS. Multiple patterns of the same type are allowed, provided that the atom (and bond) numbering corresponds between them.

## 1D*PATTERN

The PATRAN 1D patterns used in CHMTRN are similar to other linear representations such as SMILES,[17,18] SMARTS,[19] and Sybyl Line Notation,[20,21] and in fact PATRAN predates most of them. Branching is indicated using parentheses. Atoms are represented by their elemental symbols in capital letters (e.g., chlorine is “CL”), with the letter X representing any element. Bonds are represented by “-”, “=”, “#”, “%”, “&”, and “+”, for single, double, triple, aromatic, any bond type, and no bond (the separator between fragments in a disconnected graph), respectively. Alternative atoms and alternative bonds are separated by commas. Rings are indicated by an @ sign followed by the sequence number of an earlier atom in the pattern string to which a bond connects to complete the ring (atoms and bonds are automatically numbered sequentially from the left).

Atom and bond properties are inserted between square brackets. Alternative values for a property are separated by commas; alternative properties are separated by semicolons (e.g., [HS=0;ARYL=YES]). The atom properties currently supported are listed in Table 1 in the Supporting Information.

For example, the following pattern represents the substructures in Figure 1 (where no substituent is shown in Figure 1, any substituent is allowed).

$$N[HS = 1]\%C( - C[FGS = ESTER, AMIDE])\%C, N\%C\%C@1$$

Long patterns can be written on multiple lines by breaking after a bond symbol, comma, or semicolon.

## NEW*1D*PATTERN

NEW*1D*PATTERNs, which we have implemented, convey information about the precursors, as well as the products, of retro-reactions and about atom correspondences between the two sides of the retrosynthesis arrow. The representation of atoms, bonds, and their properties is the same as in 1D patterns that describe only the reaction product (target), but a retrosynthetic arrow is included and precursor patterns are written to the right of it. Within the context only of the pattern description (e.g., for use with the @ symbol), the atoms and bonds in the set of fragments to the left and right of the retrosynthetic arrow are numbered independently. Atoms to the right of the arrow are mapped to atoms on the left by appending ^ and the number of that atom, counted from the left. It is not necessary to map bonds between the two sides of the arrow since their mapping is implied by that of the atoms. Within CHMTRN statements, atoms in the set of precursors after application of the mechanism retain the numbers they had in the target; if there are new atoms or bonds in the precursors they are given sequence numbers continuing from the last number, respectively, used in the pattern string for the target. Patterns written in multiple lines can be broken following the retro-reaction arrow as well as following a bond symbol, comma, or semicolon.

For example, the following pattern, taken from transform 6026, represents the retro-reaction shown in Figure 2.

```
S[HS=0](=O)(=O)(-C[ARYL=YES])-N=>
 S^1[HS=0](=O^2)(=O^3)(-C^4[ARYL=YES])-Cl+N^5-
[HS=1,2]
```

## CHMTRN STATEMENTS

From its inception, CHMTRN was intended to be English-language like and to use chemical terms wherever appropriate.[10] A set of “buzz words” supports this. For example, the first of the following statements contains all the keywords needed by the interpreter software, but the second, making use of buzz words, is clearer to a human:

OXIME SUBTRACT 30

IF THERE IS AN OXIME ANYWHERE THEN SUBTRACT 30

The underlying purpose of CHMTRN is to manipulate sets programmatically, and this is apparent in its vocabulary and syntax. It is neither practical nor appropriate to illustrate the use of every term in the language or all of the ways in which they can be combined in a transform. The terms currently used in our SAVI project, together with brief notes about them, are described in the Supporting Information to this paper. Use of the language, and features that we have added or changed significantly, are described below.

### Control Statements.

Simple iteration can be performed by using FOR EACH in a single statement. For example

```
RAISE*RATING SLIGHTLY FOR EACH ENOLIZABLE ATOM BETA TO ATOM*2
```

Multiple sets can be selected for iteration by adding further EACH clauses, using "AND:FOR" for readability:

```
LOWER*RATING SLIGHTLY FOR EACH QUARTER-NARY*CENTER WITHIN &
GAMMA TO ATOM*2 AND:FOR EACH ALKYL*ATOM ALPHA TO ATOM*7
```

The use of DO and ENDDO in combination with FOR EACH allows multiple statements to be executed. For example

```
FOR EACH IODINE ATOM ANYWHERE DO I_CHK
    IF THERE IS ONLY ONE ATOM ALPHA TO SPECIFIED*ATOM 1  &
      AND:IF THERE IS AN AROMATIC ATOM ALPHA  &
      TO SPECIFIED*ATOM 1 THEN KILL
I_CHK   ENDDO
```

The terms " SPECIFIED*ATOM( S) n" and "SPECIFIED*BOND(S) n" are used to identify parameters passed to a subroutine (see the Supporting Information). As there is only one parameter passed to a DO loop in each iteration it is always SPECIFIED*ATOM 1 or SPECIFIED*BOND 1.

FOR EACH…DO loops can be nested to multiple levels.

The terms, LOOP and JUMP, are used in the original CHMTRN language in contexts that are not relevant to SAVI and they have not currently been reimplemented. For backward compatibility, the terms, BRANCH and THEN*TO (e.g., "BRANCH TO BR_MECH THEN*TO CL_MECH") are still used only within mechanism statements (*q.v.*), although our implementation allows them to be used anywhere.

If an END*TRANSFORM statement is encountered in a branch, the retro-reaction defined by the branch is stored and displayed, and the program moves to the next branch.

### Conditional Statements.

Basic conditional statements take the form of a single statement in either of two formats

yyy IF xxx
or
IF xxx THEN yyy
in which xxx is a single condition and yyy is a single outcome.
For example
LOWER*RATING STRONGLY IF THERE IS A KETONE ALPHA TO ATOM*3
or
IF THERE IS A KETONE ALPHA TO ATOM*3 THEN LOWER*RATING STRONGLY

Depending on the design of the CHMTRN interpreter, there may be restrictions on the use of the first of these options: if the same bits in a computer word are used to store different information depending on context, “THEN” is needed to trigger separation of the statement into separate words.

Alternative conditions, and alternative second terms in a condition, can be included in a single statement using “OR:”, “AND:”, “OR:IF”, and “AND:IF”. “OTHERWISE”, as included in the following example, can be used only if it leads to an outcome of GOTO, EXIT, or RETURN.

```
IF ON ATOM*3 THERE IS A KETONE OR: NITRILE AND:IF BOND*5  &
IS NOT A DOUBLE*BOND THEN LOWER*RATING STRONGLY  &
OTHERWISE GOTO DB_CHK
```

The first part of a statement like, “IF ON ATOM*3 THERE IS A KETONE OR: NITRILE…”, is interpreted as meaning “IF ON ATOM*3 THERE IS A KETONE OR:IF ON ATOM*3 THERE IS A NITRILE…” Multiple instances of either AND:IF or OR:IF in the same statement are supported but the use of both in the same statement is not, to avoid ambiguity (the use of parentheses in statements of this kind is not supported).

The term “AND*THEN” allows multiple outcomes to be triggered by the same condition within a single statement. For example

```
IF THERE IS A NITRILE ALPHA TO ATOM*3 THEN  &
LOWER*RATING STRONGLY AND*THEN  &
WARNING 32: Evolution of hydrogen cyanide is likely.
```

Multiple statements can be applied conditionally by using the terms BEGIN and BLKEND. For example in which the use of the label, BADSIDEGRP, must be unique within the file containing the transform.

```
IF THERE IS A CARBON ATOM ALPHA TO ATOM*2 OFFPATH
BEGIN BADSIDEGRP
    SAVE AS 1 THE CARBON ATOM ALPHA TO ATOM*2 OFFPATH
    KILL IF THERE IS MORE THAN ONE HETERO ATOM ALPHA  &
      TO SAVED*ATOM 1
BLKEND BADSIDEGRP
```

Processing within a block can be terminated, to continue with the next statement after the block end, by using EXIT followed by the label that identifies the block end. For example

```
EXIT BADSIDEGRP
```

### Mechanism Statements.

The inclusion of precursor patterns and atom-to-atom correspondences in NEW*1-D*PATTERNs might allow the CACTVS CHMTRN compiler to use transforms retrosynthetically or forward-synthetically without the need for mechanism statements (see the Supporting Information). However, they are needed for backward compatibility with LHASA and there are additional issues such as stereospecificity and handling of post mechanism statements that currently depend on them. So the NEW*1-D*PATTERNs are used for processing in the synthetic direction but the CHMTRN mechanism statements are still used for retrosynthetic processing.

### Rating.

As described in the Supporting Information, CHMTRN includes qualitative RATING statements about TYPICAL*YIELD, RELIABILITY, etc. We have added the following two factors:

**COST—**Whether expensive reagents or costly conditions, such as use of high temperature and pressure, are needed.

**SAFETY—**Whether hazardous reagents or conditions are involved.

### Qualitative Rating Adjustments.

Although a scheme for qualitative assignment of initial rating was implemented for LHASA, increments and decrements within the body of transforms remained numerical. We have introduced an alternative, qualitative categorical scheme, allowing all rating assessments to be under the control of the application that uses the transform knowledge base, where it can be done by reasoning about the relationships between categories or arithmetically.

The words RAISE*RATING and LOWER*RATING have been introduced along with terms to express the degree of change that is required: SLIGHTLY, MODERATELY, STRONGLY, or SEVERELY. For example:

IF THERE IS AN ALCOHOL OFFPATH THEN LOWER*RATING STRONGLY

RAISE*RATING SLIGHTLY IF BOND*2 IS IN A RING OF SIZE 5 THROUGH 7

For backward compatibility, numerical rating increments and decrements are still being written into transforms as well. For example, the above statement about lowering rating if there is an alcohol offpath might be accompanied by

IF THERE IS AN ALCOHOL OFFPATH THEN SUBTRACT 30

Guidelines for deciding on the degree of change to make to a rating have not yet been fully developed and there were previously no clear guidelines for deciding on the size of numerical increments or decrements. The broad advice on numerical increments and decrements was to take into consideration the size of the initial numerical rating for the transform, whereas the new scheme requires only a judgment about the change in itself—scaling of the change to suit the initial, categorical rating, where appropriate, is handled by the program (and hence potentially the end user) rather than by the transform writer. So there is no formal mapping between numerical and categorical rating changes. For the purposes of updating old transforms to the new scheme, after looking at how numerical rating changes (which were always in multiples of five) had been used in practice, we have applied the following conversions of numerical to categorical values: 5−15, SLIGHTLY; 20−25, MODERATELY; 30−35, STRONGLY; ≥40, SEVERELY.

Although, as mentioned above, our transforms currently contain rating statements of both kinds, a CHMTRN interpreter should use only one rating scheme or the other, of course.

## EXAMPLE TRANSFORM

Figure 3 illustrates the use of CHMTRN, with embedded PATRAN, in a transform. The retro-reaction diagram following the references in the transform is a representation for the convenience of anyone working on the transform − it is ignored by the compiler. As mentioned above, there are statements to modify the rating both numerically (SUBTRACT 40), for backward compatibility, and qualitatively (LOWER*RATING SEVERELY). The mechanism statements encode the separation of $R^1$ from the nitrogen atom and the attachment of a $B(OH)_2$ group to $R^1$.

## CONCLUSION

We have been using the revised CHMTRN and PATRAN for CACTVS-assisted generation of compound libraries[6] based on a preliminary set of 53 widely applicable transforms.[9] Our hope is that reporting their use, together with this application note, will encourage the wider adoption of the languages.

## SOFTWARE AVAILABILITY

A parser and execution engine for transforms written in the languages CHMTRN/PATRAN have been implemented in the chemoinformatics toolkit CACTVS. An academic version of CACTVS can be downloaded for various operating systems.[14] It is free software for

academia, nonprofits, evaluation, and for replicating published results. Commercial toolkit versions for other customers or application areas are available for licensing. Versions of CACTVS starting from version 3.4.6.13 contain all vetted transforms used in the SAVI project[6–9] in their latest versions, including the ones newly written as part of the SAVI project, as source. A publicly accessible web tool,[15] lets one upload, test, and compile CHMTRN files and download the compiled output. The SAVI database download page also provides all transforms used in that project for free download.[22]

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## ACKNOWLEDGMENTS

We thank Alan Long and Alex Sukharevsky for providing the SAVI project with approximately 1000 of the original transforms of the LHASA project and with documentation about the CHMTRN language and thank LHASA Limited for providing us with the transforms developed by their members. We thank Martin Ott for helpful advice and for writing transform 2875 (Copper[I]-catalyzed azide−alkyne cycloaddition) for us.

### Funding

Authors P.N.J., H.P., V.D., N.T., and M.C.N. received funding from the NCI, NIH, Intramural Research Program. Author W- D.I. received funding from Xemistry GmbH internal research budget.

## REFERENCES

(1). Polishchuk P; Madzhidov T; Varnek A Estimation of the Size of Drug-like Chemical Space Based on GDB-17 Data. J. Comput.-Aided Mol. Des. 2013, 27, 675–9. [PubMed: 23963658]

(2). Hoffmann T; Gastreich M The next level in chemical space navigation: going far beyond enumerable compound libraries. Drug Discovery Today 2019, 24 (5), 1148–1156. [PubMed: 30851414]

(3). https://enamine.net/library-synthesis/real-compounds (accessed May 14th 2020).

(4). https://www.biosolveit.de/REALSpaceNavigator/download/REAL_Space_Navigator_01-2018.pdf (accessed May 14th 2020).

(5). https://www.biosolveit.de/REALSpaceNavigator/ (accessed May 14th 2020).

(6). https://chemrxiv.org/articles/Synthetically_Accessible_Virtual_Inventory_SAVI_/12185559 DOI: 10.26434/chemrxiv.12185559.v1 (accessed June 9th 2020).

(7). Pevzner Y; Ihlenfeldt W-D; Nicklaus M Synthetically Accessible Virtual Inventory (SAVI). National Meeting of the American-Chemical-Society (ACS), Boston, MA, Aug. 16th, 2015. Abs. Pap. Am. Chem. Soc, 2015, 250, 50; https://tpa.acs.org/abstract/acsnm250-2255702/synthetically-accessible-virtual-inventory-savi (accessed May 14th 2020).

(8). Patel H; Ihlenfeldt W-D; Nicklaus M Analysis of Billions of Synthetically Accessible Virtual Inventory (SAVI) compounds as to their drug potential. National Meeting of the American-Chemical-Society (ACS), Orlando, FL, Mar. 31−Apr. 04, 2019. Abs. Pap. Am. Chem. Soc, 2019, 257, 1; https://tpa.acs.org/abstract/acsnm257-3111613/analysis-of-billions-of-synthetically-accessible-virtual-inventory-savi-compounds-as-to-their-drug-potential (accessed May 14th 2020).

(9). https://cactus.nci.nih.gov/download/savi_download/ (accessed May 14th 2020).

(10). Pensak D; Corey E LHASA−Logic and Heuristics Applied to Synthetic Analysis. In Computer-Assisted Organic Synthesis; ACS Symposium Series; Wipke, Howe, Eds.; American Chemical Society, 1977; Vol 61, pp 1–32;

(11). Corey E; Long A; Rubenstein S Computer-Assisted Analysis in Organic Synthesis. Science 1985, 228, 408−418. [PubMed: 3838594]

(12). Ihlenfeldt W-D; Takahashi Y; Abe H; Sasaki S Computation and Management of Chemical Properties in CACTVS: An Extensible Networked Approach Toward Modularity and Compatibility. J. Chem. Inf. Model. 1994, 34 (1), 109−116.

(13). Ihlenfeldt W-D Ushering the Cactvs Toolkit into the Python Age (without breaking the legacy). J. Cheminf 2014, 6 (S1), 34.

(14). https://www.xemistry.com/academic/ (accessed May 14th 2020).

(15). https://www.xemistry.com/ChmtrnEditor/ (accessed May 23rd 2020).

(16). Hopkinson G Computer-assisted Organic Synthesis Design. Ph.D. thesis, University of Leeds, England, 1985.

(17). Weininger D SMILES. A Chemical Language and Information System. J. Chem. Inf. Model. 1988, 28, 31–37.

(18). Weininger D; Weininger A; Weininger JL SMILES 2. A Chemical Language and Information System. J. Chem. Inf. Model. 1989, 29, 97–101.

(19). Bone R; Firth M; Sykes R SMILES Extensions for Pattern Matching and Molecular Transformations: Applications in Chemoinformatics. J. Chem. Inf. Comput. Sci. 1999, 39, 846–860.

(20). Ash S; Cline M; Homer R; Hurst T; Smith G SYBYL Line Notation (SLN): A Versatile Language for Chemical Structure Representation. J. Chem. Inf. Comput. Sci. 1997, 37, 71–79.

(21). Homer R; Swanson J; Jilek R; Hurst T; Clark R SYBYL Line Notation (SLN): A Single Notation to Represent Chemical Structures, Queries, Reactions, and Virtual Libraries. J. Chem. Inf. Model. 2008, 48, 2294–2307. [PubMed: 18998666]

(22). https://cactus.nci.nih.gov/download/savi_download/savi_2020_transforms_src_and_clb.tar (accessed June 9th 2020).

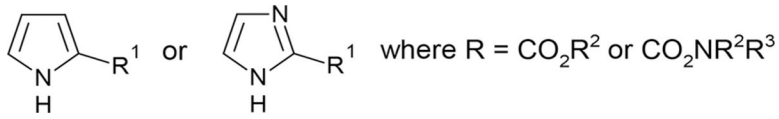



**Figure 1.**
Pair of related generic structures.

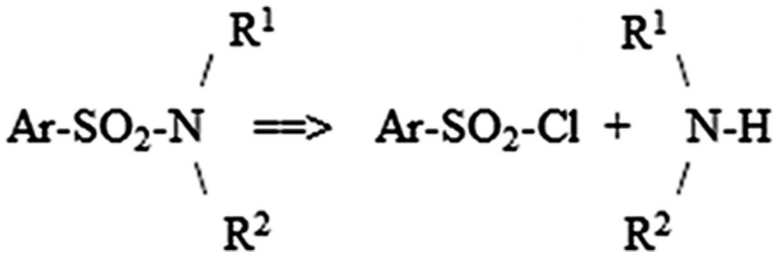



**Figure 2.**
Retro-reaction covered by transform 6026


**Figure 3.**
Example of a transform.