



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE QUIMICA

SINTESIS ORGANICA ASISTIDA POR COMPUTADORA: DESARROLLO DE UN PROGRAMA EDUCATIVO

T E S I S
QUE PARA OBTENER EL TITULO DE:
Q U I M I C O
P R E S E N T A :
LUIS IVAN TUBERT BROHMAN



MEXICO, D. F.

2001

Handwritten notes and stamps at the bottom of the page.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

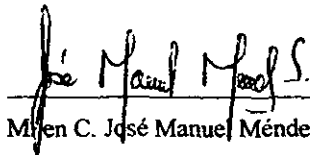
El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO

Presidente Prof. Helio Flores Ramirez
Vocal Prof. Carlos Rius Alonso
Secretario Prof. José Manuel Méndez Stivalet
Primer suplente Prof. José Gustavo Ávila Zárraga
Segundo suplente Prof. Martha Menes Arzate

Esta tesis fue desarrollada en el departamento de Química Orgánica de la Facultad de Química de la UNAM.

Asesor



M. en C. José Manuel Méndez Stivalet

Sustentante



Luis Iván Tubert Brohman

AGRADECIMIENTOS Y/O DEDICATORIA

A mi chiquita: Fabiola.

A mis padres: Juan y Marilú; y a mis hermanos: Eduardo, Mara y Daniel.

A los amigos de la carrera: Edgar, Sergio, Fabiola (¡otra vez!).

A los viejos amigos: Christian, Panchisco, Santiago, Rodrigo y Enrique.

A los olímpicos: Mauricio, Ramiro, Toña, José Manuel, Fernando, Tina, Octavio, Gloria, Mayte, Victor, Carmen, Renata, Mauricio, Alán y Carol.

A los habitantes del laboratorio 216: Laura, Rosana, Sigfrido, Edgar (otra vez) y Fabiola (¡otra vez!).

A Vicente Talanquer.

A Victor, a Toño y a el_oso, por haberme prestado computadoras.

Al Programa de Becas para Tesis de Licenciatura (PROBETEL), por su apoyo.

Al pueblo de México, por haber pagado todo con sus impuestos.

A Ronald Zollman, director de la OFUNAM.

A los estudiantes de química orgánica caídos en cumplimiento del deber.

CONTENIDO

PREFACIO	VIII
Estructura del documento	ix
1. INTRODUCCIÓN	1
2. PRINCIPIOS DE SÍNTESIS	8
2.1 Breve historia de la síntesis orgánica	8
2.2 Análisis retrosintético	13
Reacciones vs transformadas (nomenclatura)	14
Tipos de transformadas	16
Árbol retrosintético	19
Estrategias	20
Basadas en transformadas	21
Basadas en materias primas	21
Topológicas	22
Estereoquímicas	23
De grupos funcionales	23
3. PROGRAMAS DE CAOS	24
3.1 OCSS (1969)	24
3.2 LHASA (1971)	26
3.3 SYNCHEM (1973)	30
3.4 SynGen (1981)	31
3.5 CAMEO (1980)	34
3.6 WODCA (1990)	36
4. PRINCIPIOS DE CAOS	38
4.1 Operaciones básicas	38
Interfase con el usuario	38
	IV

Representación de las estructuras	39
Percepción	42
Grupos funcionales	43
Anillos	44
Canonización	44
Representación de las transformadas	46
Representación algorítmica	46
Representación abstracta	47
4.2 Operaciones avanzadas	48
Tácticas	48
Estrategias	49
5. FUNCIONAMIENTO DEL PROGRAMA OSET	51
5.1 Esquema general	52
5.2 Cliente	53
Dibujo de la estructura	53
Navegación del árbol	54
5.3 Servidor	56
Percepción	56
Anillos	57
Canonización	61
Búsqueda en la base de datos de materias primas	62
Clasificación de átomos	63
Identificación de grupos funcionales	64
Cálculo de complejidad	65
Ejecución	66
Selección del modo de análisis	66
Identificación de retronos	68
Aplicación de la transformada	69
Postprocesamiento	70
6. CASOS ESTUDIADOS	73
6.1 MarcianoI	73
Ruta 1: Reducción de Birch y alquilación de enolatos	73
Ruta 2: Adición de Michael	75
Ruta 3. Acilación de Friedel–Crafts y reducción de Birch	75
Ruta 4. Reducción de Birch y reacción de Wittig	77

Ruta 5: Diels–Alder	78
Ruta 6: Anillación de Robinson	79
6.2 Juvabiona	80
Sintesis de Ayyar y Rao (1967)	80
Sintesis de Mori y Matsui (1968)	82
7. APLICACIONES EDUCATIVAS	84
7.1 Antecedentes históricos	84
7.2 Posibles aplicaciones del programa OSET en enseñanza	87
7.3 Pruebas de campo	89
8. CONCLUSIONES	91
A. MANUAL DEL USUARIO	93
A.1 Upper panel (tree panel)	93
A.2 Lower panel (precursors panel)	94
A.3 Right panel (information panel)	95
Up button	95
Find CAS button	95
Mode selector	95
B. MANUAL DE PROGRAMACIÓN	97
B.1 Functional group description language	97
B.2 Transform Description Language	100
General notes	100
Header	101
Statements	105
Assignment	105
Branching	105
Atom and bond manipulation	106
Output	107
Functions	108
Expressions and Operators	108
Atom-type operators	108
Numerical comparison operators	109
Boolean (logical) operators	109
Math operators	109

Types and variables	109
A1-A9, B1-B9, P1-P9, R1-R9, S1-S9	110
Rating	110
C. ÍNDICE DE TRANSFORMADAS	111
REFERENCIAS	119
ÍNDICE	126

PREFACIO

Este proyecto surgió de mi interés en la programación de computadoras para aplicaciones no numéricas. Alguna vez me había preguntado si sería posible hacer un programa que “supiera” química orgánica: que manejara reacciones o mecanismos. El esquema de la química orgánica (vulgarmente llamado de “palitos y bolitas”) es ciertamente una aplicación no numérica.

Un día, paseando por los pasillos de la biblioteca de la Facultad, me topé con un artículo de Barone (1986) donde se explica con gran detalle el desarrollo de la Síntesis Orgánica Asistida por Computadora (CAOS). Entonces decidí que quería hacer mi propio programa de CAOS, principalmente por el reto que representa y porque ese tipo de pasatiempos entra en mi peculiar concepto de diversión.

Escribir un programa tan complejo y con 25,000 líneas de código es una tarea titánica que difícilmente podría haber realizado solo. Afortunadamente conté con el apoyo de mi hermano, Eduardo Tubert Brohman, quien actualmente estudia una maestría en Ciencias de la Computación en el ITESM. Decidimos establecer un proyecto conjunto, que él utilizaría para sus estudios de computación y yo para mis estudios de química.

La división del trabajo fue necesariamente heterogénea: yo me enfoqué principalmente en la parte química del programa (las partes del programa que manejan átomos y moléculas), mientras que Eduardo se enfocó a los problemas computacionales más avanzados (el intérprete del lenguaje, la comunicación en red, los problemas de teoría de grafos). Yo escribí todas las reacciones de la base de datos, ejecuté las pruebas, hice un ensayo del programa con alumnos, y busqué y leí los artículos y libros sobre síntesis orgánica.

Estructura del documento

- **1. Introducción.** Planteamiento del problema y objetivos.
- **2. Principios de síntesis.** Historia de la síntesis orgánica y del análisis retrosintético. Estrategias usadas en el diseño de síntesis.
- **3. Programas de CAOS.** Revisión histórica de los principales programas de Síntesis Orgánica Asistida por Computadora.
- **4. Principios de CAOS.** Revisión general del funcionamiento de los programas de CAOS: problemas que hay que resolver y algoritmos.
- **5. Funcionamiento del programa OSET.** Detalles acerca de la estructura interna y funcionamiento del programa desarrollado.
- **6. Casos estudiados.** Ejemplos y discusión de moléculas analizadas con OSET.
- **7. Aplicaciones educativas.** Antecedentes del uso educativo que se le ha dado a programas de CAOS. Propuestas de uso del programa OSET en la enseñanza de la síntesis orgánica, dentro del marco del plan de estudios de la carrera de Química de la Facultad de Química de la UNAM. Resultados de una prueba de OSET con alumnos de la Facultad.
- **8. Conclusiones.**
- **Apéndice A – Manual del usuario.** Explica el uso de la interfase gráfica del programa.
- **Apéndice B – Manual de programación.** Dirigido a quienes quieran introducir o modificar reacciones y grupos funcionales en la base de datos del programa.
- **Apéndice C – Índice de transformadas.** Lista de transformadas presentes en la base de datos de OSET.

1. INTRODUCCIÓN

La Química crea su objeto. Esta facultad creadora, semejante a la del arte, la distingue claramente de las ciencias naturales e históricas.

Berthelot, 1860.

¿Qué es la química? Olvidando por un momento las definiciones de los libros de texto, podría decirse que es la ciencia de crear nuevas sustancias. Además, es una ciencia muy prolífica: según Schummer (1997 y 1999), tomando en cuenta el número de publicaciones, la química es la más activa de las ciencias modernas. En la Figura 1-1 se muestra el número de *abstracts* publicados en química, física y biología.

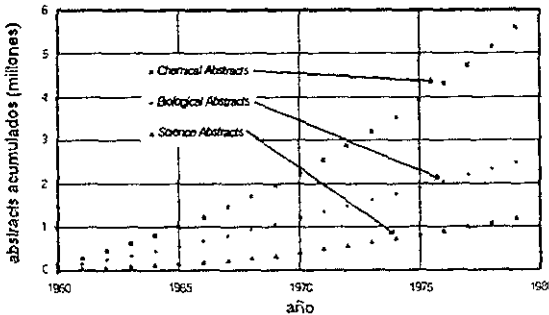


Figura 1-1 Número de *abstracts* con respecto al tiempo para diversas ciencias. Tomada de Schummer, 1997.

Se ha encontrado que la mayoría de los artículos de química publicados presentan síntesis de compuestos nuevos. Con ello el número de compuestos conocidos (la mayoría de ellos artificiales) se ha incrementado exponencialmente desde los inicios de la química moderna a la fecha, y no muestra indicios de desaceleración. Esta tendencia se muestra en la Figura 1-2.

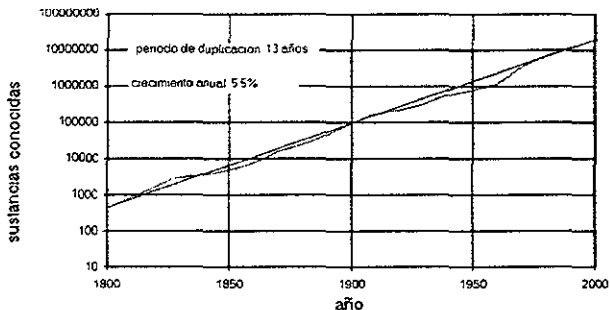


Figura 1-2 Gráfica semilogarítmica del número de sustancias conocidas con respecto al tiempo. La línea recta es la regresión lineal. Tomada de Schummer, 1997.

Una pregunta obligada es: ¿para qué sintetizan los químicos estos compuestos? En la Figura 1-3 se clasifican en cinco tipos los trabajos de síntesis publicados entre 1980 y 1995 en la afamada revista *Angewandte Chemie*. Las columnas de *teoría* y *clasificación* son las que se relacionan más fácilmente con la idea de “ciencia pura”. La categoría de *características estructurales* incluye moléculas “exóticas” como los polihedranos, catenanos y fenestranos, y su justificación es difícil de entender para quienes no son químicos (en cierta manera recuerda la situación de la síntesis de nuevos elementos). La columna de *aplicaciones* es la que se relaciona de manera directa con la tecnología y el uso práctico de sustancias en farmacia y en la industria. Llama la atención el hecho de que solamente la cuarta parte de las sustancias nuevas son sintetizadas con intención de una aplicación práctica.

Queda entonces la columna más grande, que abarca más del 40% del total: *síntesis*

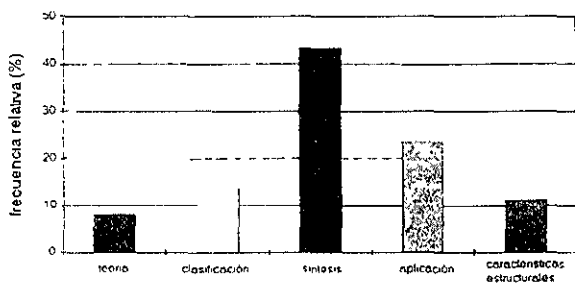


Figura 1-3 Objetivos de artículos sintéticos publicados en *Angewandte Chemie* entre 1980 y 1995. Tomada de Schummer, 1997.

Esta clase incluye las sustancias producidas con la intención de aumentar el poder sintético de la química: catalizadores, reactivos, y pruebas de métodos nuevos. Esto sugiere que la síntesis es un fin en sí mismo. Aquí es donde entra en juego la cita de Berthelot: la química estudia sustancias químicas, pero dado que puede crearlas, crea su propio objeto de estudio. Como consecuencia, la química no puede aspirar a estar completa, ya que la cantidad de sustancias posibles es prácticamente infinita.

Dado que los químicos dedican tanto esfuerzo y energía a la síntesis de nuevos compuestos, resulta natural el desarrollo de herramientas para ayudar en esta labor. Una herramienta introducida en el siglo XX fue la computadora, máquina capaz de hacer millones de cálculos laboriosos en cuestión de segundos. Aunque el diseño de síntesis no es un problema numérico, las computadoras también pueden usarse para la resolución de problemas simbólicos. El primer intento de aplicar las computadoras en el diseño de síntesis orgánica fue a fines de la década de los sesenta (Corey, 1969).

La síntesis orgánica ha sido comparada muchas veces con un juego de ajedrez. Roald Hoffmann, en su libro *Lo mismo y no lo mismo* (1995), escribió lo siguiente en un capítulo acerca de la síntesis del cubano:

La estrategia de la síntesis tiene una lógica contundente. El esquema de una síntesis de múltiples pasos se parece al planteamiento de un problema de ajedrez. Al final está el cubano: el jaque mate. En el interín hay jugadas, con reglas que las determinan. Las reglas son mucho más interesantes y flexibles que en el ajedrez. El problema del químico dedicado a la síntesis es planear un movimiento en el tablero diez jugadas antes, que tenga una apariencia de lo más común. Pero desde esa posición, un jugador (o un equipo de químicos), mediante una ingeniosa serie de jugadas, logra el jaque mate sin importar lo que haga el más recaictrante y formidable oponente de todos: la naturaleza.

Ott y Noordik (1992) construyeron la Tabla 1-1 para mostrar algunas analogías entre la síntesis orgánica y el ajedrez. Además mencionan la analogía que existe entre los sacrificios y el aumento temporal de complejidad en el análisis retrosintético (por ejemplo, debido al uso de grupos protectores). Otra analogía es la que existe entre las *combinaciones* en el ajedrez y las *combinaciones tácticas* en síntesis orgánica.

Por otra parte, Hippe (1992) comparó el *espacio fase* de la síntesis orgánica con el del ajedrez. Con esto se refiere al número de posiciones posibles en el ajedrez, comparado el número de estructuras moleculares posibles. Para un límite arbitrario de 100 átomos, se estima que el espacio fase es de 10^{120} , mientras que el espacio fase del ajedrez se estima en

	Diseño de síntesis	Ajedrez
arreglo	estructura molecular	posición
objeto	átomo	pieza
paso	reacción	jugada
restricciones	conocimiento químico	reglas
parámetros de evaluación	complejidad molecular	balance material y posicional
dirección del análisis	hacia menor complejidad	hacia mayor ventaja material y posicional
meta	compuestos disponibles	jaque mate

Tabla 1-1 Comparación entre el diseño de síntesis y el ajedrez. Tomada de Ott, 1992.

10^{80} (magnitud similar al número estimado de átomos en el universo). ¡Desde este punto de vista, el problema de la síntesis orgánica es 10^{40} veces más complejo que el del ajedrez! La conclusión es que no se puede usar la *fuerza bruta*¹ para resolver un problema de síntesis orgánica. Incluso en el ajedrez resulta imposible aplicar la fuerza bruta más allá de unas pocas jugadas, y sólo en años recientes (1997) ha sido posible que una computadora derrote al campeón mundial de ajedrez.²

Podemos ver entonces el diseño de síntesis orgánica como una especie de juego de ajedrez con muchas reglas vagamente definidas, en un tablero de tamaño indeterminado pero grande. ¿Cómo puede aspirar una computadora a resolver tal problema? ¿O incluso un humano? Los humanos, afortunadamente, poseen una intuición y flexibilidad que las computadoras no tienen. Ningún humano resuelve este tipo de problemas por fuerza bruta: aprovecha su capacidad de visión global y de comprensión. Las computadoras tienen las ventajas de rapidez de cálculo y memoria perfecta. Estas comparaciones, adaptadas al caso de la síntesis orgánica, aparecen en la Tabla 1-2, tomada de un artículo de Barone (1986).

Si no se puede realizar una búsqueda exhaustiva, es necesario tomar decisiones. Para ello se utilizan las estrategias que mencionaré en el Capítulo 2. Un término muy utilizado en todos los campos que tratan con problemas complejos, es el de *heurística*. La *heurística*, conocida desde la antigüedad como “el arte de inventar”, tradicionalmente ha

¹ Término usado en ciencias de la computación para referirse a algoritmos que consideran exhaustivamente todas las posibilidades para encontrar la mejor solución.

² Ver por ejemplo, <http://www.sciam.com/explorations/042197chess/050597chesscom.html>.

Lo que hace mejor el químico	Lo que hace mejor la computadora
Reconoce el objetivo en términos generales. Tipo de problema, dificultad, etc.	Percibe sistemáticamente las situaciones estructurales específicas. No pasa por alto nada.
Reconoce que una reacción que usualmente se considera de bajo rendimiento puede mejorarse para este caso específico	Recuerda una lista muy grande de reacciones. Un químico con una buena biblioteca puede hacer esto bien, aunque no tan rápido
Es mejor para evaluar globalmente un árbol sintético.	Habilidad para hacer incansablemente el trabajo repetitivo de escribir miles de estructuras y considerar miles de posibilidades.
Habilidad para crear nuevas estrategias	Habilidad para comparar sistemáticamente una estructura dada con miles de materias primas.
Experiencia. El buen químico sabe qué reacción es posible, cuál tiene sus mañas, y cómo pueden ser modificadas. El banco de datos de un buen químico ha sido alimentado durante muchos años de leer la literatura.	Puede explorar las consecuencias de una nueva estrategia en cientos de moléculas o en compuestos comerciales cuyo precio sube o baja.
Movilidad. El químico sabe qué es lo que no sabe, y sabe cómo buscarlo.	Utiliza todos los caminos de la química, aunque a primera vista parezcan ingenuos.
Adaptación a la nueva información. El químico puede leer un artículo y extrapolarlo al problema en turno.	Su banco de datos puede mejorar continuamente si se alimenta regularmente, y nunca olvida.

Tabla 1-2 Comparación entre las habilidades de la computadora y del humano para el diseño de síntesis
Tomada de Barone, 1986

estudiado las reglas y los métodos del descubrimiento y de la invención. La heurística moderna trata de comprender el método que conduce a la solución de problemas, y en particular las operaciones mentales típicamente útiles en este proceso (Polya, 1944).

La heurística propone reglas que pueden ayudar a resolver problemas; sin embargo, hay que enfatizar que estas reglas no son infalibles y tienen excepciones. En el siguiente capítulo mencionaré algunas reglas que suelen ayudar en la planeación de una síntesis orgánica. La mayoría de los programas de síntesis orgánica por computadora usan una combinación de heurística, análisis intenso (mas no exhaustivo), e interacción con el usuario. Todo esto lo veremos con mayor detalle en el Capítulo 3

Ante el panorama aparentemente desolador de la complejidad del problema y del crecimiento exponencial de la cantidad de compuestos conocidos, hay un signo de esperanza. En 1965, Gordon E. Moore notó que el número de componentes en los

procesadores de las computadoras, y con ello su velocidad, se duplicaba cada 18 meses, aproximadamente. Este crecimiento exponencial continúa hasta la fecha, como puede verse en la Figura 1-4. A este fenómeno se le conoce como la *Ley de Moore*.³

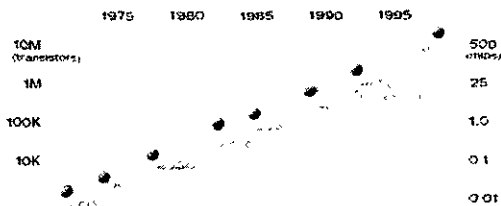


Figura 1-4 Gráfica semilogarítmica del número de transistores y rapidez de los microprocesadores con respecto al tiempo. Puede apreciarse la tendencia exponencial enunciada en la ley de Moore.

Cuando se empezó a trabajar en la síntesis orgánica asistida por computadora (CAOS), eran necesarias las computadoras más caras y sofisticadas de la época. Incluso en 1986, cuando las microcomputadoras eran todavía novedosas, quienes trabajaban en CAOS desconfiaban de su posible utilidad (Barone, 1986). Sin embargo, hoy, tal y como predijeron Barone y Chanon en su artículo de 1986, cada vez más programas están migrando hacia las microcomputadoras que se pueden encontrar en todas partes.

Desde la década de 1970 se han utilizado con fines educativos los programas de CAOS. Sin embargo, las computadoras de entonces eran poco accesibles, y por razones históricas durante los decenios siguientes los programas de CAOS seguían siendo sólo para computadoras caras. Esto, aunado a las restricciones de distribución y costo de los programas, impidió un uso masivo de este software en la enseñanza (para más detalles, ver el Capítulo 7).

El problema educativo que se intenta resolver es el siguiente: el estudiante de química orgánica a nivel licenciatura frecuentemente se siente desconcertado ante el aparente laberinto de variaciones estructurales y reacciones iconexas, las cuales intenta memorizar sin mucho éxito (Turro, 1986). Además, es común que no vea el porqué del estudio de dichas reacciones, es decir, el objetivo de la química orgánica. De acuerdo con lo

³ Para más información, ver la página de Intel, <http://www.intel.com/intel/museum/25anniv/11q1/moore.htm>

visto al principio de este capítulo, puede decirse que la meta principal de la química orgánica es la síntesis de nuevas moléculas.

El objetivo de este proyecto es aplicar la síntesis orgánica asistida por computadora a la enseñanza de la química orgánica. Para ello se desarrolló un programa que sirva como herramienta de exploración en el análisis retrosintético de estructuras moderadamente complejas. Este programa deberá ser utilizable en los sistemas más populares (Windows, MacOS, Linux, Unix) y aprovechar el internet para hacerlo accesible a la mayor cantidad posible de usuarios. El programa será *software libre*,⁴ lo que significa que estará disponible sin costo alguno junto con el código fuente para que los usuarios tengan la libertad de usarlo y modificarlo si así lo desean.

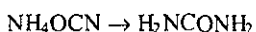
⁴ Para mayor información sobre el software libre, ver <http://www.gnu.org/>

2. PRINCIPIOS DE SÍNTESIS

En este capítulo revisaré los principios fundamentales del análisis retrosintético. Algunos de estos principios son la base del programa descrito en esta tesis, así como de otros programas anteriores.

2.1 Breve historia de la síntesis orgánica

Aunque la síntesis orgánica empírica tiene sus orígenes en la antigüedad, puede decirse que la primera síntesis famosa en la historia de la Química moderna fue la síntesis de la urea a partir del cianato de amonio realizada por Whöler en 1828.¹



La síntesis como manipulación de moléculas orgánicas para la obtención de moléculas complejas o no disponibles a partir de moléculas simples o disponibles cobró mayor importancia en los últimos cuatro decenios del siglo XIX, debido en gran medida a la teoría estructural propuesta independientemente por Couper y Kekulé en 1859 (Brown, 1959). Una de las síntesis más famosas de esta época es la del indigo (Figura 2-1), tinte de origen vegetal muypreciado. El púrpura de Tiro, tinte que en la antigüedad costaba más de diez veces su peso en oro y que estuvo reservado para el uso exclusivo del emperador de Roma, es un derivado dibromado del indigo. Para extraer un gramo de esta sustancia se requerían más de 10,000 individuos de cierta especie de moluscos del mediterráneo (Smit, 1998; Pozharskii, 1996).²

¹ Comúnmente se le atribuye a este evento la caída del pensamiento vitalista en la química orgánica. Sin embargo, el cianato de amonio utilizado era indirectamente de origen animal, y la intención de Whöler no era derrumbar al vitalismo sino demostrar la isomería. La síntesis de ácido acético de Kolbe en 1845 a partir de carbón fue más importante como evidencia contra el vitalismo.

² Cabe mencionar que desde su descubrimiento, el indigo es barato y se utiliza para teñir ropa de uso común (por ejemplo, los pantalones de mezclilla). Este es un ejemplo de cómo la ciencia puede ser democratizadora.

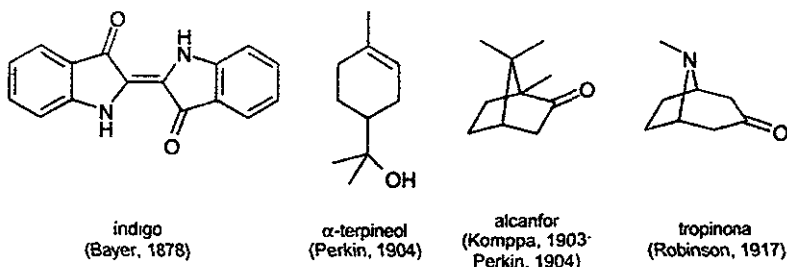


Figura 2-1 Algunas síntesis representativas de principios del siglo XX y fines del XIX.

La mayoría de las síntesis logradas en esa época fueron de compuestos aromáticos derivados del benceno, y se realizaban muchas veces durante los estudios de exploración de la reactividad, sin mediar mucha planeación. Casi siempre se partía de un compuesto con una estructura íntimamente relacionada con el que se iba a sintetizar y se realizaban las refuncionalizaciones necesarias (Corey, 1989).

A principios del siglo XX empezaron a aparecer síntesis con una planeación más

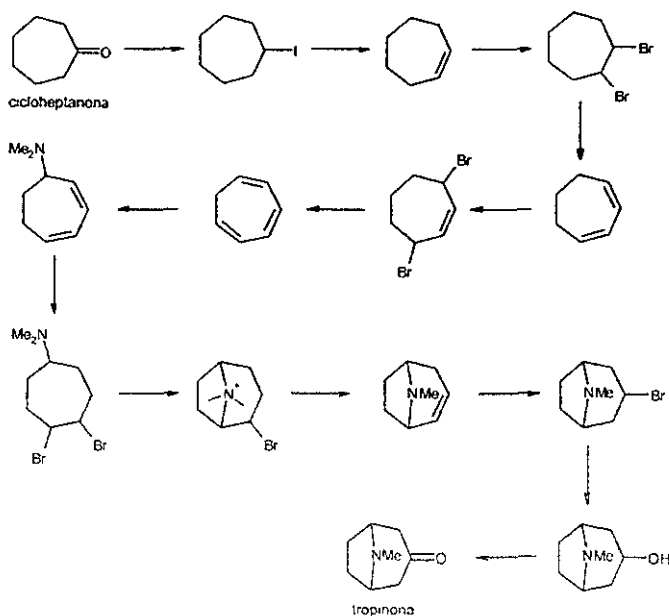


Figura 2-2 Síntesis de Willstätter de la tropinona

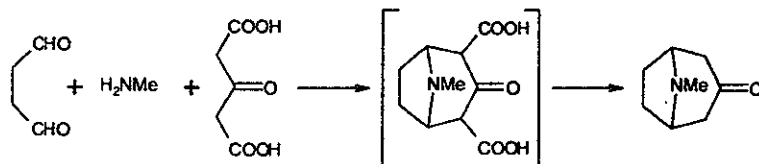
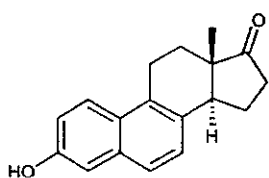


Figura 2-3 Síntesis de Robinson de la tropinona. El intermediario entre corchetes no se aísla.

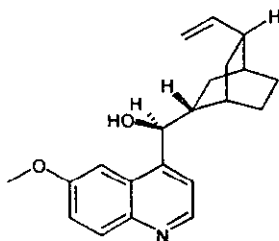
sutil y que ya estaban basadas en reacciones constructivas para formar el esqueleto de la molécula sin necesidad de utilizar materias primas relacionadas. Los ejemplos más representativos de este periodo son el α -terpineol, el alcanfor y la tropinona (Figura 2-1).

La síntesis de la tropinona es un ejemplo interesante de este cambio de paradigma en síntesis orgánica (Smit, 1998; Turner, 1976). En 1902, Willstater sintetizó la tropinona a partir de cicloheptanona, un compuesto bastante similar: contenía de antemano todos los enlaces carbono-carbono de la tropinona y el grupo funcional cetona. Sin embargo, la síntesis —mostrada en la Figura 2-2— consistió en 13 pasos y tuvo un rendimiento global de 0.75%, a pesar de que todos los pasos eran reacciones bien conocidas y con rendimientos aceptables.

En 1917, Robinson sintetizó la tropinona ¡en un solo paso! partiendo de compuestos que no tienen semejanza alguna con el producto final, utilizando la reacción de Mannich seguida por una descarboxilación *in situ*. La síntesis se muestra en Figura 2-3. El rendimiento fue de 17%, y con algunas modificaciones se pudo mejorar a más de 90% (por Schopf, en 1937). Este es uno de los mejores ejemplos de por qué no es recomendable limitar el análisis a materias primas semejantes a la molécula objetivo.



Equimelina
(Bachman, 1939)



Quinina
(Woodward, Doering, 1944)

Figura 2-4 Síntesis de la mitad del siglo XX.

Durante los años 1920-1945 la *tendencia hacia una mayor complejidad continuó*, como puede verse en las estructuras de la equinelina y la quinina, mostradas en la Figura 2-4. En esta época empezaron a realizarse síntesis estereoselectivas. Después de la segunda guerra mundial las síntesis se volvieron más complejas debido a cinco factores (Corey, 1989):

1. La formulación de mecanismos detallados para las principales reacciones orgánicas.
2. El uso del análisis conformacional de las estructuras orgánicas y de los estados de transición, el cual sirvió de base para el desarrollo de reacciones estereoselectivas.
3. El uso de los métodos espectroscópicos e instrumentales de análisis.
4. La aplicación de los métodos cromatográficos de análisis y de separación.
5. El descubrimiento y aplicación de nuevos reactivos selectivos para las reacciones orgánicas.

Algunas síntesis representativas del período 1945-1960 son la de la vitamina A, la cortisona, la reserpina y la penicilina V (Figura 2-5). Sin embargo, según Corey (1989),

Durante este período cada problema sintético era considerado como un caso especial con un análisis individualizado. El pensamiento del químico era dominado por el problema en

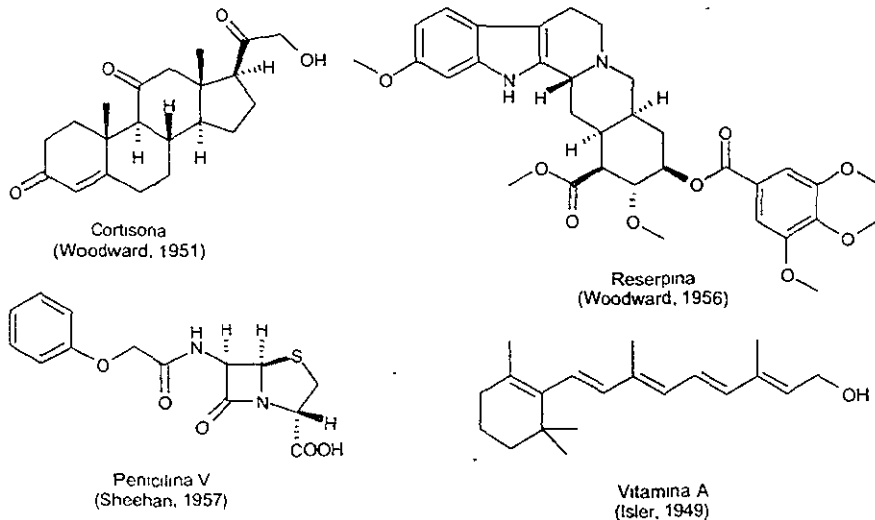


Figura 2-5 Algunas síntesis del período 1945-1960

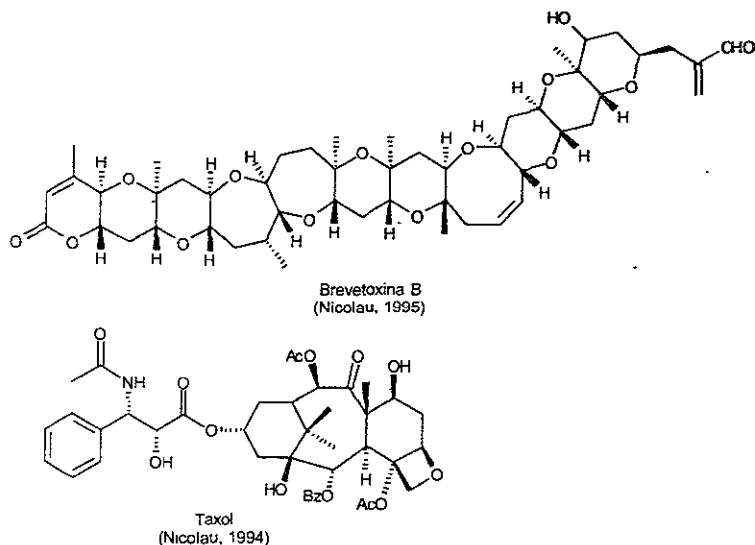


Figura 2-6 Síntesis recientes.

cuestión. Gran parte del pensamiento no estaba guiado, o era dirigido inconscientemente. Durante los 1950s y en muchas escuelas hasta en los 1970s, la síntesis era enseñada mediante la presentación de una serie de síntesis reales ilustrativas (y generalmente no relacionadas). Los químicos que aprendían síntesis mediante este “enfoque de casos” atacaban cada problema de una manera improvisada. La búsqueda intuitiva de pistas para la solución del problema no era guiada por técnicas efectivas de solución de problemas aplicadas de manera consciente

Fue en la década de los sesenta cuando se enunció de manera explícita el método retrosintético (Corey, 1967), el cual veremos con mayor detalle en la siguiente sección. El análisis retrosintético ahora se utiliza rutinariamente en la planeación de síntesis complejas. Algunos ejemplos de las moléculas más complejas sintetizadas en los últimos años se pueden en la Figura 2-6, donde se muestran las estructuras de la brevetoxina B y el taxol

¿Cuál será el siguiente período en la historia de la síntesis orgánica? Ha habido algunos desarrollos interesantes, como la síntesis con soporte sólido y la síntesis combinatoria; sin embargo, podría decirse que estos desarrollos son más bien una ramificación de la línea de desarrollo de la síntesis que he seguido en esta sección, que es la línea de la complejidad creciente. La síntesis combinatoria nos permite sintetizar moléculas con mayor rapidez, pero no necesariamente más complejas; la síntesis en estado sólido

también es aplicable para síntesis de moléculas muy complejas como las proteínas y ácidos nucleicos, pero no tiene una aplicación general en síntesis orgánica todavía.

Hay quienes creen que ya es posible sintetizar prácticamente cualquier molécula compleja, pero la síntesis orgánica todavía tiene mucho por delante. Basta con comparar las síntesis de laboratorio con las que ocurren en la naturaleza para darse una idea: no solamente es deseable poder sintetizar una molécula, sino hacerlo de la manera más suave, limpia y barata. Según Turner (1976), la síntesis ideal es una que *ocurre en un solo paso, con 100% de rendimiento a partir de materiales accesibles, en medio acuoso y condiciones suaves (temperatura ambiente, pH cercano al neutro), no tiene subproductos y es fácil de separar el producto deseado*. Definitivamente este es el extremo de la idealización, y dudo mucho que este tipo de síntesis predomine algún día. Sin embargo, es la dirección hacia la que se desea que evolucione la síntesis orgánica.

En esta sección mostré una periodización de la historia de la síntesis orgánica. Hay que tomar en cuenta que no es la única posible y que los límites entre los periodos son en cierta medida arbitrarios. Por otra parte, en esta breve cronología me basé en la síntesis de productos naturales, pero no hice mención de algunas moléculas muy interesantes que han sido grandes logros en la historia de la síntesis y que presentan retos especiales. En esta situación se encuentran el cubano, el dodecaedrano, los catenanos, fenestranos, helicenos, rotaxanos, radialenos, cumulenos, etc

2.2 Análisis retrosintético

Aunque generalmente se le atribuye a Corey la invención del análisis retrosintético,³ podría decirse *a posteriori* que lo que hizo fue simplemente aplicar a la química el método de análisis que ya era conocido por los matemáticos griegos en el siglo III a.C. Pappus, matemático griego, escribió más o menos lo siguiente (Polya, 1956. Las cursivas son mías.):

En el análisis, partiendo de lo que es requerido, lo consideramos como admitido, sacamos las consecuencias, después las consecuencias de dichas consecuencias, hasta llegar a un punto que podamos utilizar como punto de partida para una síntesis. Pues en análisis admitimos.

³ Ver por ejemplo la página de la Fundación Nobel: <http://www.nobel.se/laureates/chemistry-1990-press.html>

como ya hecho, lo que nos piden que hagamos, como encontrado lo que buscamos, como verdadero lo que hay que demostrar. *Buscamos de qué antecedente se podría deducir el resultado deseado; después buscamos cuál podría ser el antecedente de este antecedente, y así sucesivamente, hasta que llegando de un antecedente a otro, encontremos finalmente alguna cosa conocida o admitida como cierta.* Dicho proceso lo llamamos análisis, solución hacia atrás o razonamiento regresivo.

Si donde dice *antecedente* leemos *precursor*, donde dice *cosa conocida* leemos *materia prima disponible*, donde dice *resultado* leemos *molécula*, etc. esta descripción se parece sorprendentemente a cualquier descripción moderna del método de análisis retrosintético. A continuación cito una descripción reciente del análisis retrosintético escrita por Corey (1989):

El análisis retrosintético (o antitético) es una técnica de resolución de problemas para la transformación de una molécula objetivo sintético (MOB) en una secuencia de estructuras cada vez más simples a lo largo de un camino que lleva finalmente a materias primas simples o comercialmente disponibles para la síntesis química. La transformación de una molécula en un precursor sintético es realizada mediante la aplicación de una transformada, el inverso exacto de una reacción química, a la molécula objetivo. Cada estructura derivada antitéticamente de una MOB se convierte entonces en una nueva MOB para analizar. La repetición de este proceso produce finalmente un árbol de intermediarios que tiene estructuras químicas como nodos y donde los caminos de abajo hacia arriba corresponden a posibles rutas sintéticas de la MOB.

En la cita anterior aparecen varios de los términos más importantes del análisis retrosintético. En las secciones que siguen los definiré con más detalle.

Reacciones vs transformadas (nomenclatura)

En la Tabla 2-1 se comparan los términos comúnmente usados en el análisis retrosintético con sus homólogos en la dirección sintética.

Uno de los conceptos fundamentales en el análisis retrosintético es el de transformada. Una transformada es la operación mental exactamente inversa de la reacción en el sentido sintético. Las transformadas toman el mismo nombre de la reacción correspondiente: por ejemplo, la transformada correspondiente a la reacción de condensación aldólica es la transformada de condensación aldólica (Figura 2-7 a y b). No hay que confundir las transformadas con las reacciones inversas que existen para algunas reacciones

	dirección sintética	dirección retrosintética
paso	reacción	transformada
flecha	\longrightarrow	\Longrightarrow
estructura de partida (cola de la flecha)	reactivo	objetivo
estructura resultante (punta de la flecha)	producto	precursor
subestructura requerida para la operación	grupos funcionales	retrón

Tabla 2-1 Diferencias entre la notación sintética y la retrosintética.

reversibles: por ejemplo, existe la retro-condensación aldólica (figura 2-7 c), la retro-Diels–Alder, etc. Estas reacciones inversas son reacciones propiamente dichas, con sus propias condiciones de reacción y limitaciones, mientras que las transformadas son mentales y existen para todas las reacciones (incluyendo las reacciones inversas). Como puede verse en la figura, las transformadas se representan con una flecha doble que apunta del objetivo a los precursores, es decir, en la dirección del análisis.

La subestructura que al estar presente en un objetivo permite la aplicación de una transformada se conoce como *retrón*. El retrón puede ser un grupo funcional, dos grupos unidos por una cadena de cierta longitud, uno o más anillos o algún otro tipo de patrón no clasificable en las categorías anteriores. En el ejemplo de la condensación aldólica, el retrón consiste en un alcohol y un aldehído separados por un camino de dos enlaces C–C. En la dirección sintética generalmente se requieren ciertos grupos funcionales en los reactivos para poder aplicar una reacción: en el caso de la condensación aldólica se necesitan dos

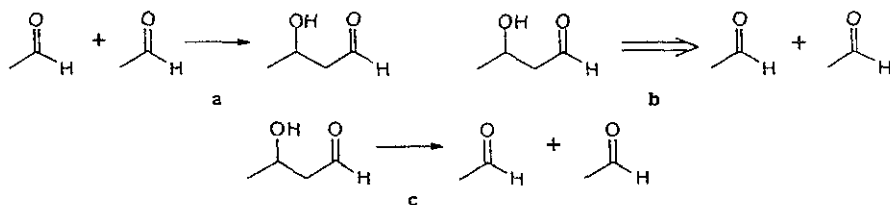


Figura 2-7 a) Ejemplo de reacción de condensación aldólica, b) transformada de condensación aldólica; c) reacción de retro-condensación aldólica

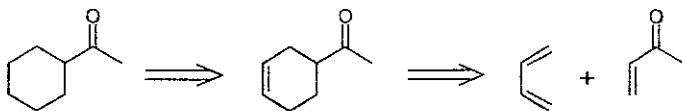


Figura 2-8 Ejemplo de un retrón parcial. La ciclohexil metil cetona tiene un retrón parcial para la transformada de Diels-Alder, el cual se completa mediante la transformada de hidrogenación catalítica

aldehidos.

Una molécula objetivo también puede tener *retrones parciales*, los cuales sugieren la aplicación de una transformada; pero, para poder aplicarla, es necesario emplear transformadas auxiliares que permitan formar el retrón completo. Por ejemplo, en la figura se analiza la ciclohexil metil cetona, que contiene parte del retrón para la transformada de Diels-Alder. El retrón completo consiste en un anillo de seis átomos de carbono con un doble enlace y un grupo electroatractor. Si a nuestra molécula objetivo le aplicamos la transformada de hidrogenación catalítica de dobles enlaces, podemos agregar un doble enlace en la posición requerida para tener el retrón completo y aplicar posteriormente la transformada de Diels-Alder.

Como puede verse si se examina cualquier síntesis compleja, las transformadas simplificadoras son menos utilizadas que las transformadas auxiliares (Hendrickson, 1998). Para Hendrickson, una síntesis ideal sería aquella donde todas las transformadas utilizadas fueran simplificadoras. Sin embargo, la realidad es diferente: para poder aplicar las transformadas simplificadoras suele ser necesario modificar grupos funcionales para obtener los retronos necesarios; además, es común que haya que colocar y remover grupos protectores, desenmascarar grupos funcionales, etc

Tipos de transformadas

Existen diversas clasificaciones para las reacciones orgánicas. La más usada se basa en el mecanismo, dividiendo las reacciones en clases como sustitución, adición, eliminación, radicales libres, pericíclicas, oxidaciones y reducciones. Este es el esquema usado en diversos libros de texto, como el de March (1992). Claro está que hay reacciones que pueden caer en más de una categoría (especialmente si su mecanismo es complejo), o hay reacciones con mecanismos poco conocidos o con mecanismos que varían según las

condiciones. En muchos de estos casos se prefiere clasificar una reacción de acuerdo con el cambio estructural neto más que por su mecanismo. A pesar de todo, esta clasificación ha funcionado bastante bien y es muy utilizada.

También han sido propuestas otras clasificaciones más rigurosas desde un punto de vista formal y más exhaustivas (Hendrickson, 1992). Estas clasificaciones se basan en general en el cambio neto ya que el mecanismo puede ser un criterio incierto. En el artículo de Hendrickson de 1992 se mencionan tres clasificaciones que, según las conclusiones del autor, son formalmente equivalentes. A continuación describiré brevemente las tres clasificaciones.

La de Fujita (Hendrickson, 1992), se basa en *estados de transición imaginarios*, donde se clasifican los enlaces en formados, rotos y no alterados. Hay que enfatizar que este estado de transición imaginario representa el cambio estructural neto y generalmente no tiene nada que ver con el mecanismo de la reacción.

La clasificación de Dugundji-Ugi (Dugundji, 1976) se basa en la representación de la reacción como una matriz que se suma a la *matriz de enlace-electrón* del conjunto de reactivos. Existe una matriz diferente para cada caso particular de cada reacción, pero los autores encontraron que todas estas matrices podían ser reducidas (simplificadas) a un número finito y manejable. Con sólo 30 matrices reducidas es posible representar prácticamente todas las reacciones conocidas y desconocidas.

El método de Hendrickson (1990, 1992) se basa en la división de la reacción en dos semirreacciones: una "electrofílica" y una "nucleofílica". Nuevamente, estos términos no implica nada acerca del mecanismo, sino que se basan en el cambio neto de algunas propiedades selectas de los átomos involucrados, como el número de oxidación y el número de enlaces σ y π . Con este método también es posible clasificar reacciones conocidas y desconocidas, y tiene la ventaja de que es el que más se parece al utilizado tradicionalmente por los químicos.

Es interesante mencionar que dos de estas clasificaciones formales han estado relacionadas con proyectos computacionales: tanto Ugi como Hendrickson han creado programas de CAOS, como veremos en el siguiente capítulo.

La clasificación clásica por mecanismo que mencioné al principio de esta sección se utiliza principalmente para clasificar reacciones, no transformadas, y para las

clasificaciones formales no hay gran diferencia entre reacción y transformada, ya que el cambio neto puede con igual facilidad “sumársele” a los reactivos o “restársele” a los productos (en el caso del método de matrices, éstas literalmente se suman y se restan).

Hay otra clasificación informal que está diseñada especialmente para el análisis retrosintético y que ha sido usada en programas como LHASA (Corey, 1989). Esta es básicamente la clasificación que usamos en esta tesis.

Las transformadas se clasifican como simplificadoras o auxiliares. En el análisis de una molécula se consideran como directamente útiles solamente las transformadas simplificadoras. Las transformadas auxiliares son vistas como medios para poder utilizar las transformadas simplificadoras. En segundo lugar las transformadas se clasifican por el tipo de retrón o por el tipo de operación realizada.

Las transformadas simplificadoras pueden clasificarse por su tipo de retrón en:

- **Transformadas de un grupo funcional.** Ejemplos: alquilación de enolatos, esterificación, reacción de Wittig.
- **Transformadas de dos grupos funcionales.** Ejemplos: condensación aldólica, reacción de Mannich, adición de Michael.
- **Transformadas de anillos:** Ejemplos: Diels–Alder, anillación de Robinson, formación de ciclopropanos.
- **Remoción de grupos funcionales.** Este es un tipo de transformada relativamente poco común, que por su tipo de retrón puede clasificarse como de un grupo funcional (ejemplo: halogenación por radicales libres) o de dos grupos funcionales (ejemplo: halogenación alfa a carbonilo). Sin embargo la incluyo aquí explícitamente por si el lector se pregunta al leer sobre la categoría de adición de grupos funcionales si no existirá también la de remoción de grupos funcionales. Otra característica que diferencia a este tipo de transformada de las demás transformadas simplificadoras es que no desconecta enlaces esqueléticos.

Pueden existir transformadas con retrones más complejos que no se puedan clasificar en las categorías anteriores. Puede haber transformadas de n grupos funcionales, o de sistemas heterocíclicos y policíclicos complejos. En programas como LHASA se ha optado por agregar una categoría llamada *transformadas de patrón arbitrario*, donde se describe el retrón como un patrón de átomos de cualquier tamaño. En esta categoría se puede poner

cualquier transformada que no quepa en las otras. Quizá se preguntará el lector por qué no describir todas las transformadas con patrones y olvidarnos de la clasificación. La respuesta es que el reconocimiento de patrones (especialmente patrones grandes) es mucho menos eficiente desde el punto de vista computacional.

Las transformadas auxiliares se clasifican por el tipo de operación auxiliar realizada:

- **Intercambio de grupos funcionales (FGI).** Este es el tipo de transformada con más ejemplos, ya que prácticamente es posible convertir cualquier grupo funcional en cualquier otro, especialmente si ambos grupos funcionales se encuentran en el mismo nivel de oxidación⁴ (Smit, 1998). Generalmente, hay muchas opciones de intercambios de grupos funcionales para cualquier molécula objetivo, por lo que solamente conviene usar estas transformadas cuando se tiene algún objetivo en mente, como puede ser la aplicación de una transformada simplificativa.
- **Adición de grupos funcionales (FGA).** Esta es una adición en el sentido retrosintético y corresponde a una remoción en el sentido sintético. Como ejemplos podemos mencionar la reducción de Clemmensen/Wolff-Kishner, la reducción de halogenuros de alquilo a alcanos y la hidrogenación catalítica de alquenos y alquinos. Este tipo de transformada, aun más que la de intercambio de grupos funcionales, sólo debe usarse cuando hay alguna razón específica, ya que las transformadas de FGA pueden aplicarse casi en cualquier parte de la molécula, dando una avalancha de precursores probablemente inútiles. Además, la adición de un nuevo grupo funcional vuelve más compleja la molécula, por lo que va en contra de la dirección deseada del análisis

Árbol retrosintético

Si se aplican diversas transformadas a una molécula objetivo se obtiene una lista de precursores. Si a cada precursor se le considera a su vez como una nueva MOB y se analiza, se obtiene una lista de precursores para cada precursor. La estructura resultante es lo que en teoría de grafos se conoce como árbol

⁴ A las reacciones donde se conserva el estado de oxidación se les conoce como isohipsicas

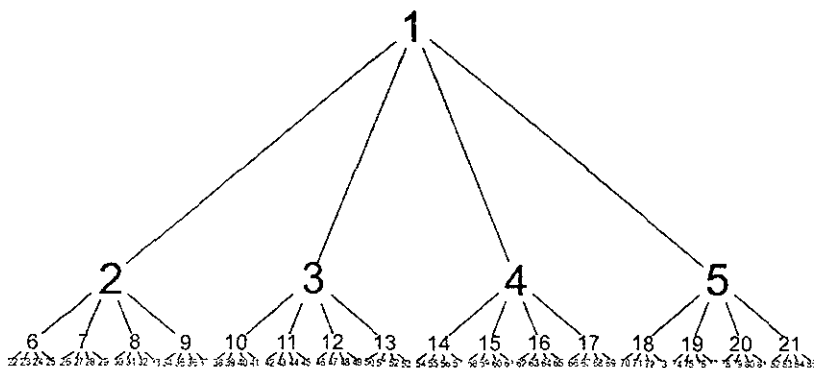


Figura 2-9 Ejemplo de árbol que crece exponencialmente como 4^n .

Si se expande el árbol sintético a varios niveles de profundidad, se observa que crece de manera exponencial. Por ejemplo, si cada paso de análisis genera 10 precursores, se encuentra que el tamaño del nivel n del árbol tiene 10^n estructuras. ¡Si analizamos exhaustivamente a diez pasos de profundidad obtenemos 10,000,000,000 de precursores! En la Figura 2-9 se muestra de manera gráfica esta situación idealizada para cuatro precursores por paso de análisis.

Estrategias

La esencia del análisis retrosintético es la generación del árbol retrosintético de la molécula objetivo mediante la aplicación reiterada de transformadas. Se suele decir que este análisis debe ser tan exhaustivo como sea posible para asegurarse de encontrar las buenas rutas sintéticas; sin embargo, debido al crecimiento exponencial del árbol, resulta imposible realizar un análisis verdaderamente exhaustivo. La conclusión es que *necesariamente hay que seleccionar las transformadas que se deben utilizar*. De ello se encargan la estrategias.

Las estrategias se pueden clasificar en las siguientes clases (Corey, 1989). 1) basadas en transformadas; 2) basadas en materias primas; 3) topológicas; 4) estereoquímicas; y 5) basadas en grupos funcionales. La estrategia óptima depende en gran medida de la molécula que se esté analizando, pero generalmente es conveniente y necesario aplicar una combinación de distintas estrategias.

Basadas en transformadas

El objetivo del análisis retrosintético es la simplificación de la molécula. Por ello, en general es mejor una transformada que simplifique la molécula que una que la deja con igual o mayor complejidad. Sin embargo, la mayoría de las transformadas posibles *no son simplificativas*. La esencia de las estrategias basadas en transformadas es, entonces, utilizar preferentemente las transformadas simplificativas.

Sin embargo, no es posible utilizar exclusivamente transformadas simplificativas, aun cuando ello sería lo ideal; es necesario utilizar transformadas auxiliares que nos permitan la simplificación. Existe una cantidad relativamente grande de transformadas simplificativas, así que es necesario restringir la búsqueda aun más. Lo lógico es utilizar las transformadas más poderosas, es decir, aquellas que sean muy simplificativas, de aplicación general y con buenos rendimientos.

Existe una cantidad relativamente pequeña de transformadas consideradas como poderosas. Entre ellas se encuentran la de Diels–Alder, la anillación de Robinson, la reducción de Birch, y diversas ciclizaciones y transposiciones (Corey, 1985a). Para realizar el análisis utilizando esta estrategia, el químico selecciona primero la transformada que quiere utilizar tomando en cuenta las características estructurales de la molécula. A esta transformada se le llama transformada objetivo (*goal transform*). El análisis consiste entonces en la aplicación de transformadas auxiliares o subobjetivos (*subgoals*), con el fin de obtener el retrón necesario para la transformada objetivo.

Basadas en materias primas

Si se conocen materias primas disponibles semejantes a la molécula objetivo, éstas pueden servir de guía. El análisis se concentra en las transformadas que aumenten la semejanza con las materias primas; además, es posible hacer también el análisis en la dirección sintética o, mejor aun, en ambas direcciones simultáneamente. Este es el tipo de análisis más viejo y, como vimos con el ejemplo de la tropinona, no siempre es el más apropiado. Sin embargo todavía sigue vigente y en algunos casos sí es la mejor opción.

Los casos donde resulta más obvio el uso este tipo de estrategia son aquellos en los que la molécula está conformada por “ladrillos” fácilmente ensamblables. En este caso se encuentran los péptidos, o moléculas como la *buspirona* (Figura 2-10).

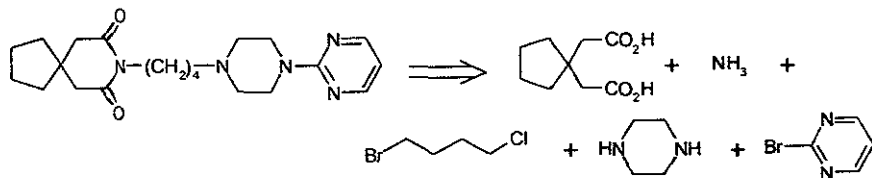


Figura 2-10 Desconexión de la buspirona en “ladrillos” constituyentes.

Topológicas

Uno de los elementos más importantes de complejidad estructural es la *complejidad topológica*. Esta se debe a la presencia de anillos y ramificaciones en la molécula. Corey ha enunciado de manera explícita conjuntos de reglas heurísticas para identificar *enlaces estratégicos* en diversos tipos de estructuras: acíclicas, cíclicas aisladas, policíclicas fusionadas, puenteadas y espirofusionadas (Corey, 1989). Un enlace estratégico es aquel que al ser desconectado produce una gran simplificación estructural. Esto es especialmente notable en el caso de sistemas puenteados complejos: puede verse en la Figura 2-11 que la desconexión de cualquiera de los enlaces estratégicos (marcados con un trazo grueso) reduce el sistema a uno fusionado mucho más simple. Además de los enlaces estratégicos, se definen enlaces *preservados*, es decir aquellos que *no* deben desconectarse en el análisis.

Las estrategias topológicas se encuentran entre las más útiles en el análisis retrosintético y, usadas en conjunto con otras estrategias, pueden redundar en los siguientes beneficios (Corey, 1989):

1. Disminución del tiempo requerido para encontrar soluciones excelentes.
2. Descubrimiento de rutas convergentes o especialmente cortas
3. Control efectivo de la estereoquímica.
4. Regioselectividad.
5. Minimización de problemas de reactividad.



Figura 2-11 Ejemplo de enlaces estratégicos en un sistema puenteado. Si se desconecta cualquiera de los enlaces marcados, se obtiene una estructura fusionada simple

6. Facilitación de pasos químicos cruciales.

Estereoquímicas

Las estrategias estereoquímicas facilitan la remoción de estereocentros y la simplificación de las relaciones entre ellos bajo condiciones de estereocontrol. Dicho control puede estar basado en la transformada (por ejemplo, con reacciones como la epoxidación asimétrica de Sharpless), o en el propio sustrato. La estereoselectividad inducida por el sustrato depende de la presencia de otros estereocentros que orienten una reacción que normalmente no sería selectiva. En esta categoría se incluye una gran cantidad de reacciones estereoselectivas, como la reducción de carbonilos y la condensación aldólica, pero todas ellas dependen de la estereoquímica preexistente en el precursor (en algunos casos es posible introducirla temporalmente con grupos auxiliares).

La finalidad de las estrategias estereoquímicas es, con base en este conocimiento de las transformadas asimétricas, determinar cuáles son los estereocentros que se deben remover durante el análisis retrosintético y en qué orden, y cuáles se deben conservar.

De grupos funcionales

La presencia de grupos funcionales en la molécula puede indicar el uso de ciertas transformadas simplificadoras; de hecho, la mayoría de las transformadas contienen uno o dos grupos funcionales en su retrón. A veces se le llama *estrategia oportunista* a la aplicación de transformadas solamente porque la molécula objetivo ya tiene los grupos funcionales requeridos.

Hay transformadas simplificadoras que eliminan grupos funcionales (es decir, los reemplazan por hidrógeno). Además, el intercambio de grupos funcionales (FGI) es una táctica comúnmente usada como parte de este tipo de estrategia. Las estrategias de grupos funcionales se utilizan casi siempre asociadas con otras estrategias, como las topológicas y las estereoquímicas.

3. PROGRAMAS DE CAOS

No pretendo hacer una revisión exhaustiva de todos los programas de CAOS que existen y han existido; para ello, existe una excelente revisión de 1986 que menciona 46 programas (Barone, 1986),¹ así como algunas revisiones más recientes pero menos completas (Ihlenfeldt, 1995; Ugi, 1993; Ott, 1992) y otra más antigua (Bersohn, 1976). Más bien pretendo analizar algunos de los programas que considero más representativos o que más influencia tuvieron en esta tesis.

El año indicado entre paréntesis en cada uno de los títulos siguientes es el año de la primera publicación acerca del programa en cuestión.

3.1 OCSS (1969)

El programa OCSS (*Organic Chemistry Simulation of Synthesis*) fue el primer programa de CAOS en ser publicado (Corey, 1969). Este programa fue desarrollado en Harvard por el grupo de Corey, con base en los principios de análisis retrosintético publicados anteriormente por el propio Corey (1967). Aunque todo parece indicar que OCSS ya no se utiliza, habiendo sido desplazado por LHASA, es importante mencionarlo por ser el primer esfuerzo en el campo de la síntesis orgánica asistida por computadora, y en el cual se han basado muchos otros.

OCSS corría en una DEC PDP-1, y contaba con una interfase gráfica. Las moléculas eran dibujadas por el químico mediante una *tableta Rand*, y los resultados se veían en monitor o se imprimían con un *plotter*. El programa operaba de manera interactiva: una vez que el químico dibujaba la estructura y escogía la estrategia, OCSS la analizaba,

¹ Hay que tomar en cuenta que no todos estos programas son independientes sino que pueden ser versiones diferentes, y que algunos pertenecían a proyectos ya abandonados. Por ejemplo, aparecen en la lista tanto el SYNCHEM como el SYNCHEM2. Además, la amplitud y tamaño de los distintos proyectos varía enormemente.

presentando una lista de precursores. El usuario escogía entonces algún precursor para analizar, y así sucesivamente hasta que consideraba suficientemente explorado el árbol sintético.

El programa podía manipular estructuras hasta con 32 átomos explícitos, de los elementos C, H, O, N, S, P, y X (halógeno genérico). Los átomos podían tener carga o ser radicales. Aunque permitía dibujar las moléculas con estereoquímica, en realidad no la tomaba en cuenta durante el análisis. Las estructuras eran representadas internamente por una tabla de átomos y una tabla de enlaces.

OCSS percibía los grupos funcionales mediante una serie de tablas entrelazadas. Los anillos eran percibidos mediante un algoritmo típico de búsqueda por profundidad, y el programa usaba el algoritmo de canonización de Morgan (1965) para facilitar la comparación de estructuras y detectar la simetría topológica. La referencia no aclara el uso dado por el programa a la simetría percibida (quizá no la usaba todavía pero estaba en sus planes).

El corazón de todo programa de CAOS está en su representación de las transformadas. OCSS contemplaba dos tipos de transformadas: las de *mecanismo simbólico* y las de *modificación simbólica de grupos funcionales*. El programa no contaba con una base de datos de transformadas propiamente dicha, como otros sistemas posteriores, sino que representaba las transformadas de una manera generalizada.

Las transformadas de mecanismo simbólico consisten precisamente en un algoritmo con reglas mecanísticas, pero que no necesariamente corresponden de manera exacta con las reacciones que representan. Por ejemplo, existía un mecanismo llamado *desconexión tipo -OH*, que incluye todas las reacciones de adición a carbonilo y grupos análogos, incluyendo la adición de reactivos de Grignard a carbonilos, iminas y nitrilos y la condensación aldólica, entre otras. Los otros tipos de transformadas de mecanismo simbólico mencionados en el artículo de Corey (1967) son las de *eliminación cíclica* que incluye cicloadiciones, ciclizaciones y oligomerizaciones, y las de *desconexión tipo W* que incluye reacciones como la adición de Michael y alquilación de enolatos.

Las transformadas de modificación simbólica de grupos funcionales eran usadas solamente como subobjetivos a petición expresa de una transformada de mecanismo simbólico, e incluían las transformadas actualmente clasificadas como FGI y FGA. Por

ejemplo, las reacciones de desconexión tipo -OH podían utilizar en lugar del grupo -OH un halógeno, teniendo como subobjetivo el intercambio del halógeno por el hidroxilo. Las reacciones de eliminación cíclica, que se intentaban aplicar a todos los anillos de tamaño par, podían tener como subobjetivo la adición de un doble enlace al anillo.

Para las reacciones de intercambio de grupos funcionales, los autores del OCSS decidieron suponer para simplificar que, *dada la gran cantidad de métodos sintéticos disponibles, es altamente probable que se pueda transformar cualquier grupo funcional en cualquier otro*. Es decir, no había ninguna tabla de interconversiones permitidas, ni lista de reacciones de tipo FGI. Todos los intercambios de grupos funcionales estaban permitidos.

OCSS realizaba varias evaluaciones de los precursores generados. Primero verificaba que no fuera una molécula “ilegal”; luego la comparaba con las generadas anteriormente para eliminar duplicados; posteriormente calculaba la complejidad del precursor generado con el fin de mostrar los resultados por orden decreciente de simplificación; y por último, evaluaba la situación general del nuevo precursor en el árbol. Esta última evaluación es interesante porque consideraba la existencia de caminos alternos de síntesis para el mismo precursor, lo cual en algunos casos llevaba a la eliminación del camino en consideración (cuando existe ya un camino más corto y que está contenido en el camino actual) y en otros se consideraba como una ventaja (cuando se tienen caminos alternos totalmente diferentes)

3.2 LHASA (1971)

LHASA (*Logic and Heuristics Applied to Synthetic Analysis*) es un programa derivado de OCSS y que aún sigue en desarrollo.² LHASA fue originado por el grupo de Corey pero actualmente también participan en su desarrollo el grupo LHASA-UK en la Universidad de Leeds, Reino Unido, y el CAOS/CAMM Center en los Países Bajos (Ott, 2000).

Ha habido tres versiones principales de LHASA: LHASA-1, que corría en una computadora PDP-1; LHASA-10, que corría en una PDP-10; y LHASA-11, para la DEC VAX. La versión 11 es la que se utiliza desde 1981, y hasta 1992 solamente funcionaba en computadoras VAX con el sistema operativo VMS. Actualmente el programa ha sido

² Para una revisión general sobre LHASA, ver Corey (1985a)

portado a otras máquinas con sistema UNIX, tales como la Silicon Graphics, HP, y Alpha. El programa está escrito en FORTRAN y según una referencia de 1977 (Pensak, 1977) consistía en 30,000 líneas de código,³ aunque ahora también tiene partes escritas en C (Ott, 1997). Actualmente tiene más de 250,000 líneas (según su página de internet, ubicada en <http://lhasa.harvard.edu/>).

LHASA es un programa mucho más complejo que su predecesor. Cuenta con una base de datos con más de 2100 transformadas y 500 combinaciones tácticas (Rozendaal,

```

TRANSFORM 117
NAME MICHAEL ADDITION OF HETERO NUCLEOPHILE
...HET-C-C-W => HET-H + C=C-W
...MARCH 585; HOUSE 596; B+P 468
...ORG. RXNS. VOL.5 79-135 (1949)
...BULL. SOC. CHEM. FR. 254,325 (1962)
...PATH 2 BONDS
RATING 50 ...Old Rating 40
GROUP*1 MUST BE KETONE OR CYANO OR ESTER OR ACID
      OR LACTONE OR AMIDE*3 OR AMIDE*2 OR AMIDE*1
      OR LACTAM OR VINYLIW OR ALDEHYDE
GROUP 2 MUST BE ETHER OR AMINE*1 OR AMINE*2 OR AMINE*3
      OR SULPHIDE OR THIOL
STUDENT
REMOVES*STEREO CARBON2*1 ATOM*2
BROKEN*BONDS BOND2*1
...
KILL IF NO HYDROGEN ON ATOM*2
      ...REQUIRED FOR REACTION
KILL IF MULTIPLE BOND ON ATOM*2 OFFPATH OR: ON ATOM*3 OFFPATH
      .. WOULD PRODUCE ALLENIC PRECURSOR
IF BOND2*1 IS NOT IN A RING OF SIZE 5 THROUGH 7 &
      THEN KILL IF BOND2*1 IS IN A RING
SUBSTRACT 15 IF LEAVING GROUP ON ATOM*3
      ...POSSIBLE ELIMINATION
ADD 15 IF ANOTHER WITHDRAWING BOND ON ATOM*2
      ...EASIER ADDITION
SUBSTRACT 15 FOR EACH WITHDRAWING BOND ON ATOM*3
      ...UNDESIREED MICHAEL POSSIBLE
SUBSTRACT 10 IF ATOM*3 IS A TERTIARY*CENTER
IF NOT OLEFIN ON BOND*2 THEN KILL IF ATOM*2 IS NOT ENOLIZABLE
      ...STABLE ENOL PROVIDES DRIVING FORCE
IF SECOND GROUP IS ETHER THEN CONDITIONS NaOR
IF SECOND GROUP IS AMINE THEN CONDITIONS RNH2
IF SECOND GROUP IS SULPHIDE OR: THIOL THEN &
      CONDITIONS NaSR
...
      BREAK BOND2*1
      JOIN ATOM*2 AND ATOM*3
....

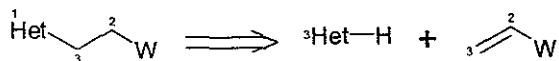
```

Figura 3-1 Ejemplo de transformada de LHASA escrita en CHMTRN Las líneas que comienzan con tres puntos () son comentarios Tomado de Corey, 1985a

³ Nuestro programa, OSET, actualmente consta de 25,000 líneas de código, en C y Java principalmente

1994). Toma en cuenta la estereoquímica, considera las condiciones de reacción y los problemas de selectividad. Puede sugerir grupos protectores (Corey, 1985b) y utiliza diversas estrategias, como las descritas en *The Logic of Chemical Synthesis* (Corey, 1989).

Las transformadas utilizadas por LHASA están escritas en un lenguaje especialmente diseñado llamado CHMTRN (*Chemistry Translator*), muy parecido al inglés. Un ejemplo de este lenguaje puede verse en la Figura 3-1. El lenguaje parece ser bastante fácil de leer, aunque no tan fácil de escribir, debido a la gran cantidad de palabras clave utilizadas. La numeración de los átomos no parece muy obvia, pero después de un análisis cuidadoso del código se puede concluir que la numeración es la siguiente:



En general los programas de CAOS que utilizan bases de datos de transformadas necesitan reglas implícitas para numerar los átomos de los retones. En el Capítulo 5 describiré el enfoque utilizado en el programa OSET para resolver este problema.

LHASA utiliza diversas estrategias. La selección de la estrategia a utilizar es realizada por el usuario, aunque ha habido intentos para facilitar la selección automática. Actualmente el programa cuenta con los siguientes tipos de estrategias:⁴

- **Basadas en transformadas, o de largo alcance.** Estas consisten en la realización de una búsqueda de largo alcance para lograr la utilización de una transformada especialmente poderosa. Las transformadas que han sido implementadas son las siguientes: Diels–Alder, Diels–Alder de quinonas (Ott, 1997), anillación de Robinson, reducción de Birch, halolactonización (Corey, 1980), ciclización de polienos (Boiten, 1993), ciclopropanación con diazocarbonos, y la síntesis estereoselectiva de olefinas. La búsqueda de largo alcance se basa principalmente en transformadas auxiliares de intercambio y adición de grupos funcionales, pero ya se permite también el uso de cualquier transformada como auxiliar.

⁴ Cualquier parecido con las estrategias discutidas en el capítulo anterior se debe a que son el trabajo del mismo autor

- **Basadas en grupos funcionales.** Este conjunto incluye estrategias de desenmascaramiento de grupos funcionales, desconexiones, reconexiones, y la “no-estrategia” de aplicación libre de transformadas de grupos funcionales. Esta última sólo es recomendable para moléculas muy simples.
- **Topológicas.** Esas estrategias se basan en la desconexión de *enlaces estratégicos*. Existen reglas claramente definidas que ayudan a identificar los enlaces estratégicos en sistemas acíclicos, cíclicos, bicíclicos y policíclicos (Corey, 1989). Si así lo desea, el usuario puede señalar manualmente el enlace que desea desconectar aunque no haya sido clasificado como estratégico por el programa.
- **Estereoquímicas.** Los estereocentros son un factor de complejidad molecular importante. Por ello es que al realizar el análisis retrosintético es necesario remover algunos de estos estereocentros. Las estrategias para removerlos utilizan transformadas enatio- y diastereoselectivas. El usuario debe seleccionar los estereocentros que quiere remover.
- **Basadas en materias primas** (Johnson, 1992). Una de las estrategias más antiguas que usan los químicos sintéticos consiste en buscar una materia prima disponible que sea similar a la molécula objetivo. Curiosamente, esta es de las más difíciles de programar en una computadora. Por ello es que este es el tipo de estrategia más recientemente implementada en LHASA. Según el artículo citado, la estrategia era eficiente cuando el usuario especificaba una materia prima deseada o un pequeño conjunto de materias primas posibles; todavía no era lo suficientemente rápido para utilizar bases de datos de gran tamaño.

LHASA evalúa cada transformada para medir su utilidad desde el punto de vista químico. Todas las transformadas tienen una calificación inicial (*initial rating*), que representa el rendimiento típico de la reacción, su confiabilidad, reputación, homoselectividad, heteroselectividad, selectividad de orientación, flexibilidad de las condiciones de la reacción, y termodinámica (Corey, 1992) Esta calificación fue realizada mediante una encuesta a diversos químicos sintéticos. Además, cada vez que se aplica una transformada se calcula su calificación final (*final rating*), que mide también el efecto de las

características específicas de la molécula objetivo sobre la viabilidad de la reacción. Por ejemplo, en la Figura 3-1 hay una línea que dice

```
SUBSTRACT 10 IF ATOM*3 IS A TERTIARY*CENTER;
```

esto significa que la reacción es menos viable si el átomo 3 es terciario (debido al impedimento estérico).

LHASA realiza un análisis de la compatibilidad de los diversos grupos funcionales de la molécula con las condiciones de la reacción que se está utilizando. Para ello cuenta con una lista de 138 *condiciones prototipo*, y una tabla que indica la reactividad de cada uno de los 78 grupos funcionales reconocidos en cada una de las condiciones de reacción. Si los grupos que no deben reaccionar son menos reactivos que el que sí debe, no hay problema; si son más reactivos el programa intenta estimar la factibilidad de proteger selectivamente los grupos que interfieren. Con base en este análisis, LHASA modifica la evaluación de la transformada.

Al igual que su predecesor, OCSS, el programa LHASA opera de manera interactiva. El usuario introduce la molécula, escoge la estrategia y el programa realiza el análisis. Si lo considera necesario, el usuario puede escoger uno de los precursores generados y analizarlo. La diferencia es que LHASA puede realizar búsquedas de largo alcance de hasta 25 pasos cuando se utiliza la estrategia basada en transformadas.

3.3 SYNCHEM (1973)

El programa SYNCHEM (Gelernter, 1977) es interesante porque está planeado para funcionar de manera totalmente no interactiva. Esto significa que una vez que el usuario introduce la estructura por analizar, ya no se requiere mayor intervención de su parte, excepto quizá el ajuste de algunos parámetros. El programa genera de manera automática el árbol retrosintético; para ello necesita un poder de cómputo considerable, además de buenas estrategias de búsqueda ya que casi siempre es imposible la exploración exhaustiva.

Una característica interesante de este programa es que cuenta con una extensa base de datos de materias primas. Esta base de datos era el catálogo de Aldrich, el cual en los

años 70 estaba disponible en notación lineal de Wiswesser, WLN (Gelernter, 1977). Los artículos más recientes no hacen mayor mención del origen de la base de datos.⁵

El programa SYNCHEM es el único del que tengo noticia que ha sido corrido de manera distribuida en una red de computadoras (Krebsbach, 1998). En el artículo citado se utilizó una red local con 9 estaciones de trabajo Sparc20, cada una con cuatro procesadores de 50 MHz y 64 MB de RAM. Sin embargo, creo que otra posibilidad prometedora es el uso de un gran número de computadoras personales en red, ya que son cada vez más baratas y ubicuas. Además, no es necesario que las computadoras estén dedicadas exclusivamente a este fin, ya que el programa podría correr discretamente en el fondo. Esto sería en cierta medida análogo a proyectos como el SETI@home,⁶ pero en una escala mucho menor. Parafraseando a *Sun Microsystems*: “la red es la computadora”. Como veremos en el capítulo sobre OSET, nuestro programa puede adaptarse fácilmente para funcionar en red, aunque puede haber problemas de escalamiento si la red es muy grande.

SYNCHEM no es el único programa no interactivo que existe. Otro ejemplo importante es el de Bersohn y colaboradores (Takabake, 1990; Takahashi, 1990; Dogane, 1992).

3.4 SynGen (1981)

El programa SynGen (*Synthesis Generator*), desarrollado por Hendrickson, sigue un enfoque totalmente distinto al de los programas que hemos mencionado. No cuenta con base de datos de reacciones ni utiliza grupos funcionales. SynGen utiliza un conjunto de reglas básicas generalizadas para generar las transformadas utilizadas en la síntesis. Gracias a esto, es quizá el único programa que, dentro de las restricciones que lo definen, explora de manera totalmente exhaustiva el árbol retrosintético con la meta de encontrar la *síntesis ideal*.⁷

⁵ Una comunicación personal con Aldrich reveló que ya no publican el catálogo en este formato

⁶ Proyecto que consiste en el análisis de señales de radiotelescopios con la finalidad de buscar inteligencia extraterrestre. Para mayor información ver su página web <http://setiathome.ssl.berkeley.edu/>

⁷ Según Hendrickson (1998), una síntesis ideal es aquella que consiste únicamente en reacciones *constructivas*, sin refuncionalizaciones

Su método se basa en tres principios (Hendrickson, 1990): concentrarse en el esqueleto de la molécula, generalizando los grupos funcionales para reducir los detalles innecesarios; utilizar solamente síntesis convergentes; y generar solamente síntesis con precursores contenidos en una base de datos.

En el programa SynGen se considera solamente el esqueleto de las moléculas, es decir, solamente los átomos de carbono y algunos heteroátomos esqueléticos. La información de los grupos funcionales se toma en cuenta registrando los tipos de enlaces que tiene cada átomo esquelético: enlaces σ a carbono, enlaces π a carbono, enlaces hacia átomos electronegativos, y enlaces hacia átomos electropositivos. De esta manera, un carbono de éster se vuelve equivalente a uno de nitrilo o a uno trihalogenado. El razonamiento detrás de esto es que los grupos funcionales diferentes pero con igual nivel de oxidación son fácilmente interconvertibles mediante reacciones isohípsicas (Smit, 1998).

Las reacciones son representadas como la suma de dos semirreacciones: una electrofílica y otra nucleofílica. No importa si el verdadero mecanismo de la reacción es diferente, por ejemplo por radicales libres; la clasificación es formal y solamente toma en cuenta el cambio neto en los tipos de átomos de la estructura. Para clasificar las reacciones constructivas solamente se necesitan 25 semirreacciones: 16 nucleofílicas y 9 electrofílicas, a partir de las cuales se pueden construir 114 tipos de reacción diferentes (Hendrickson, 1989).

Como parte del enfoque centrado en el esqueleto, SynGen decide primero desde un punto de vista puramente topológico los enlaces por desconectar y posteriormente hace el análisis químico que consiste en aplicar las semirreacciones necesarias. El análisis de desconexiones tiene solamente dos pasos de profundidad y consiste en partir la estructura aproximadamente a la mitad y luego en "cuartas partes". Cada corte consiste en la desconexión de un enlace acíclico o dos enlaces cíclicos. Para obtener solamente síntesis convergentes, es necesario que cada corte divida la molécula más o menos a la mitad; el criterio tomado en la práctica es que el fragmento más pequeño tenga por lo menos la cuarta parte de los átomos de la estructura.

Una vez que se tienen los cuatro precursores, se buscan en una base de datos con más de 6000 compuestos. Hay que recordar que tanto la estructura que se está buscando

como las estructuras de la base de datos están generalizadas en cuanto a funcionalidad. Así, por ejemplo, el 1-bromo-1-clorociclopentano sería equivalente a la ciclopentanona.

Debido a la enorme generalidad de su método de descripción de las reacciones y a que considera una gran cantidad de desconexiones, SynGen generaría un árbol sintético enorme y que contendría muchas reacciones con dudosa validez química. Para evitar esto y reducir el árbol a un tamaño manejable es necesario restringir el espacio de búsqueda mediante ciertos parámetros. Uno de ellos es la búsqueda de precursores en la base de datos. Las síntesis que parten de precursores que no se encuentran en la base de datos son eliminadas. Además, se realiza una “evaluación química” de cada transformada.

La evaluación química realizada después del análisis topológico tiene dos razones muy importantes: primero que nada, es necesario conocer el tipo de funcionalidad de los precursores para permitir su búsqueda en la base de datos; y segundo, es en esta etapa donde se eliminan las transformadas “prohibidas” y se marcan otras como “sospechosas” para que el usuario las analice. Para ello existen unas tablas de tipos de átomos necesarios o prohibidos en los alrededores del centro de reacción (Hendrickson, 1989).

El enfoque de Hendrickson es tan general que puede usarse no solamente para el análisis retrosintético, sino también para el sintético. Para ello desarrolló el programa FORWARD (palabra que significa “hacia delante” en inglés; Hendrickson, 1992). Este programa opera inicialmente de la misma manera que SynGen, partiendo la estructura en cuatro pedazos y buscándolos en la base de datos. Pero una vez que tiene los cuatro precursores, FORWARD les aplica todas las reacciones posibles en la dirección sintética. Para evitar desviarse demasiado de la ruta sintética correcta, solamente guarda las reacciones que acerquen los precursores al objetivo, mediante la evaluación de su *distancia química*. La utilidad principal del programa FORWARD es que no solamente propone síntesis para la molécula objetivo inicial, sino también para toda una serie de moléculas relacionadas. Esto puede tener dos ventajas: primero, los derivados generados muchas veces pueden transformarse fácilmente en la molécula objetivo; y segundo, en industrias como la farmacéutica suele desarrollarse la síntesis de moléculas similares a otras conocidas.

La principal ventaja de SynGen, su generalidad, es también su principal debilidad. El programa propone en muchas ocasiones transformadas que no se conocen pero que

pueden parecer razonables químicamente. Sin embargo, también propone algunas transformadas completamente absurdas (Hendrickson, 1989). Como se mencionó arriba, SynGen tiene mecanismos de selección para mitigar este problema, pero son como un arma de doble filo: existe por un lado el riesgo de eliminar accidentalmente una transformada razonable, y por otro lado el de no eliminar absurdos.⁸ La filosofía del programa SynGen es ser relativamente laxo en este sentido para que el programa pueda proponer química nueva.

Uno de los desarrollos más recientes ha sido la integración del programa SynGen con bases de datos de reacciones tales como REACCS y SYNLIB (Hendrickson, 1998). Estas bases de datos contienen cientos de miles de reacciones específicas, con datos de rendimiento y referencias. La búsqueda de reacciones análogas a la que se desea realizar siempre es útil para el químico sintético.

3.5 CAMEO (1980)

El programa CAMEO (*Computer-Assisted Mechanistic Evaluation of Organic Reactions*) trata un problema completamente diferente que todos los programas mencionados anteriormente. Como su nombre lo indica, no es un programa de análisis sintético sino de *análisis mecanístico*. El problema a tratar es el siguiente: para un conjunto dado de moléculas reactantes y condiciones de reacción, predecir cuáles son los productos principales y subproductos de la reacción (Salatin, 1980).

CAMEO intenta imitar el razonamiento mecanístico de los químicos orgánicos, el llamado “empuje de electrones”. Para ello cuenta con diversos módulos correspondientes a los principales tipos de mecanismo de reacción (Jorgensen, 1990):

- nucleofílica/catalizada por base
- electrofílica/catalizado por ácido
 - reacciones electrofílicas en general
 - sustitución electrofílica aromática
- pericíclica

⁸ Resulta interesante mencionar que, según Gund (1980), los usuarios de los programas de CAOS suelen sentirse más molestos cuando el programa sugiere una transformación absurda que cuando omite alguna transformación obvia.

- oxidación/reducción
- radicales libres
- carbenos y carbenoides
- formación de heterociclos

El programa CAMEO es bastante sofisticado: como puede verse abarca la mayor parte de la química orgánica, y además toma en cuenta problemas como la estereoselectividad de las reacciones, impedimento estérico, etc. En 1990 el programa consistía en más de 100,000 líneas de código en FORTRAN y corría en computadoras VAX con el sistema operativo VMS. También se han hecho versiones para UNIX y AIX y se está desarrollando una versión para la computadora Macintosh.

Como parte de su análisis mecanístico, CAMEO realiza diversas estimaciones cuantitativas: pK_a , ΔH°_R , energías de HOMO y LUMO, disociación de enlace, etc. Este ha sido sin duda un efecto colateral benéfico de las investigaciones en síntesis orgánica por computadora.

Un programa como CAMEO es el complemento natural para los programas de análisis retrosintético, ya que con él es posible analizar los precursores generados retrosintéticamente, con el fin de evaluar posibles problemas de selectividad y subproductos (Corey, 1985a; Jorgensen, 1990). Sin embargo, hasta ahora no he podido encontrar referencias que sugieran que esta integración ha sido llevada a cabo.

El uso del programa CAMEO es bastante sencillo: el usuario traza las estructuras de los reactivos, escoge el módulo que quiere usar para el análisis y selecciona los reactivos y condiciones de reacción específicos. El programa genera entonces un árbol de intermediarios y productos para ser examinados por el usuario. Cada uno de los productos es clasificado como "principal", "menor" o "desfavorecido".

El funcionamiento interno consta de las mismas dos etapas básicas que la mayoría de los programas mencionados en esta tesis: percepción y análisis. En este programa la percepción es un poco más penetrante que en la mayoría ya que, como ya mencionamos, realiza estimaciones de pK_a , ΔH°_R , etc. Además detecta los tautómeros (Roos-Kozel, 1981), ya que estos se forman con alta frecuencia en el análisis mecanístico, situación que no ocurre en el análisis retrosintético. El análisis mecanístico consiste básicamente en la aplicación de los pasos mecanísticos básicos: transferencia de protones, adiciones,

eliminaciones, sustituciones, etc. Cada vez que aplica un paso, el programa estima la probabilidad de que dicho paso se lleve a cabo, y analiza la estructura resultante para evitar la formación de moléculas altamente inestables.

Recientemente (a principios del año 2000) apareció en el internet una versión de demostración de CAMEO en forma de *applet* y que aparentemente tiene una arquitectura cliente-servidor (<http://zarbi.chem.yale.edu/programs/cameo.html>). Esto es interesante porque ahora cualquiera puede conocer el funcionamiento de este programa fácilmente y desde cualquier computadora. Esta es la misma idea que seguimos en esta tesis al diseñar el programa OSET con un enfoque cliente-servidor y que tiene otras ventajas como veremos en el Capítulo 5. Este *applet* es bastante lento: esperamos que el nuestro sea mejor.

3.6 WODCA (1990)

Los programas de síntesis orgánica por computadora no han sido muy usados por los químicos. LHASA (Pensak, 1977), y algunos programas relacionados como SECS (Gund, 1980) y CASP, han sido usados por empresas químicas y farmacéuticas, pero hay poca información al respecto. Sin embargo, se ha visto que los químicos sintéticos generalmente prefieren usar directamente bases de datos de reacciones tales como ORAC, REACCS y SYNLIB (Ihlenfeldt, 1995). Estas bases de datos consisten en decenas o cientos de miles de casos específicos de reacciones, con información sobre las condiciones, el rendimiento y la referencia correspondiente.

Según Ihlenfeldt esto se debe a dos problemas principales: 1) es demasiado costoso mantener actualizadas las bases de datos de los programas tradicionales; y 2) los programas tradicionales son demasiado rígidos y no corresponden a la verdadera manera de trabajar de los químicos. En realidad el químico no trabaja secuencialmente en un orden rigurosamente retrosintético, sino que examina globalmente la molécula objetivo, la compara con materias primas potenciales, se imagina posibles intermediarios, etc. Solamente al final del análisis empieza a preocuparse por reacciones y moléculas específicas. Puede decirse que los *programas de primera generación* tienen limitaciones conceptuales.

Los autores ofrecen entonces un *programa de segunda generación*, llamado WODCA (*Workbench for the Organization of Data for Chemical Applications*). Más que

un programa es un paquete de programas relacionados con una interfase común. Están enfocados a los siguientes aspectos:

- **Identificación de enlaces estratégicos.** El paquete cuenta con varios algoritmos para este fin, pero además permite al usuario marcar o remover manualmente enlaces estratégicos, con lo que se logra mayor flexibilidad.
- **Búsqueda de compuestos similares.** WODCA cuenta con varios algoritmos para ello. Es posible encontrar materias primas similares a la molécula objetivo, o a una subestructura de ella. Algunos algoritmos de búsqueda generalizan las estructuras utilizando “reacciones genéricas”, con lo que se facilita la identificación de moléculas fácilmente interconvertibles. El programa cuenta con catálogos de materias primas que incluyen información sobre el precio y el proveedor.
- **Evaluación de reacciones.** Uno de los programas integrados con WODCA es EROS (*Evaluation of Reactions for Organic Synthesis*). Este programa sirve para analizar las reacciones en la dirección sintética, y mediante la evaluación de diversos parámetros fisicoquímicos, intenta predecir el rendimiento y la rapidez de la reacción.
- **Desconexiones.** WODCA puede analizar la ruptura genérica de los enlaces estratégicos, y evaluar la viabilidad de desconectar cada uno de ellos mediante un análisis de la estabilización de la carga en los precursores (de una manera muy similar al método del sintón). Los fragmentos pueden ser sometidos nuevamente a la búsqueda de moléculas semejantes y a la identificación de enlaces estratégicos

La principal característica de WODCA es que la búsqueda no se hace paso por paso sino que está muy enfocada a las materias primas, además de ser muy interactivo. Uno de los planes que tienen los autores del sistema es integrarlo con bases de datos de reacciones. Con ello se podría facilitar el acceso a la información relevante en las bases de datos y se ahorraría mucho tiempo.

4. PRINCIPIOS DE CAOS

En este capítulo describiré el funcionamiento general de un programa de CAOS, utilizando como ejemplos algunos de los programas existentes. Mencionaré los problemas que hay que tomar en cuenta para el desarrollo de dichos programas, como son la representación de estructuras y reacciones, la percepción, las estrategias, etc. El capítulo está dividido en dos secciones: operaciones básicas, que son aquellas operaciones fundamentales necesarias en casi todos los programas; y operaciones avanzadas, aquellas que no son tan generales pero suelen encontrarse en los programas más complejos. Esta clasificación de operaciones está basada en el artículo de Barone (1986).

4.1 Operaciones básicas

Interfase con el usuario

Cualquier programa de CAOS necesita una manera de comunicarse con el usuario. Desde los primeros programas (OCSS) se han utilizado interfases gráficas. Las moléculas son dibujadas por el usuario mediante el ratón, tableta Rand, o cualquier otro método. Esto parece muy obvio en la actualidad, cuando prácticamente todos los programas tienen interfase gráfica, pero en los años 60 era una novedad. En aquellos tiempos la mayoría de los programas de química que manejaban estructuras utilizaban archivos de texto que tenían que ser dolorosamente redactados y editados por el usuario.

Ahora que la mayoría de los químicos están familiarizados con las convenciones de los sistemas operativos gráficos como Windows, MacOS, X, etc., es mucho más fácil hacer programas de computadora de fácil aprendizaje. Si además consideramos que la mayoría de los químicos están familiarizados con programas de dibujo de estructuras como ChemWin, ChemSketch o ISIS/Draw, es de esperarse que cualquier químico, con pocos minutos de capacitación, pueda usar un programa de CAOS. Esa fue una de las metas perseguidas en

esta tesis. Claro está que es necesario que el usuario tenga una idea de qué es lo que está haciendo y para qué sirve el programa, porque de lo contrario no podrá hacer nada útil aunque el programa sea muy amigable.

Representación de las estructuras

Desde hace casi siglo y medio, el método preferido por los químicos para representar las moléculas orgánicas ha sido el diagrama estructural (Brown, 1959). Éste ha evolucionado bastante y se han adoptado convenciones y variaciones para representar aspectos como la estereoquímica y la conformación.

Sin embargo, para representar una estructura en una computadora es necesario adoptar una representación computacionalmente simple, no ambigua y que no dependa de la interpretación del lector. Los tipos de representación más utilizados son las notaciones lineales, las tablas de enlace y las matrices de enlace. Todas ellas están basadas en la teoría de grafos.

Desde el punto de vista matemático, las moléculas pueden ser representadas como *grafos* (Gondran, 1979). Un grafo se define como un conjunto de *nodos*, y un conjunto de *arcos*, donde cada arco es un par de nodos.¹ Los nodos y los arcos son conceptos abstractos, que pueden corresponder a objetos un poco más concretos cuando se aplica la teoría de grafos a un problema particular. En el caso de las estructuras químicas, los nodos son los átomos y los arcos son los enlaces, y como los enlaces en química son siempre

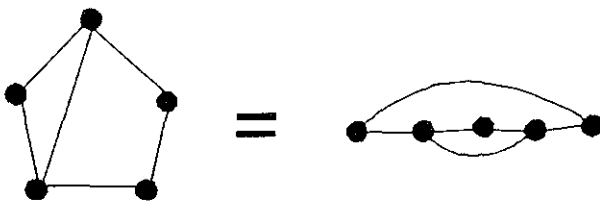
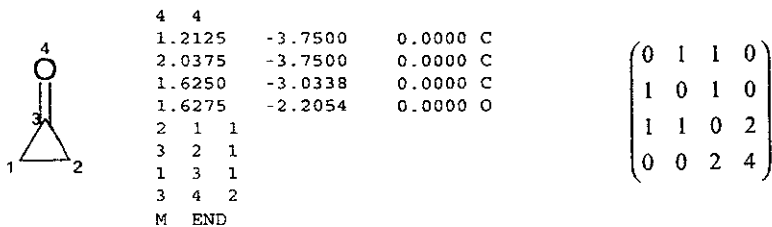


Figura 4-1 Dos grafos equivalentes. Sólo importa el número de nodos y la manera en que están conectados, la forma es irrelevante

¹ Hay otras nomenclaturas en teoría de grafos una de ellas utiliza los términos *vértice* y *arista* en lugar de nodo y arco.



Ciclopropanona

MOLFILE (simplificado)

Matriz BE

Figura 4-2 Representación de la ciclopropanona como tabla de enlace tipo MOLFILE y como matriz enlace-electrón (BE).

bidireccionales, se trata de un grafo *no dirigido*. Además, dado que hay diversos tipos de átomos (elementos) y tipos (órdenes) de enlace, se dice que es un grafo *coloreado*.

La limitación de los grafos es que los nodos no tienen posición espacial, solamente tienen otros nodos enlazados. Desde este punto de vista, los dos grafos de la Figura 4-1 son estrictamente equivalentes. La consecuencia de ello es que desde el punto de vista gráfico, no se toman en cuenta la estereoquímica, la conformación ni el impedimento estérico.

Se han hecho adaptaciones del modelo estrictamente gráfico para incluir estos aspectos; sin embargo, los primeros programas de CAOS sí utilizaban el modelo gráfico. OSET utiliza el modelo de la molécula como grafo, ya que desde un principio se decidió que la estereoquímica quedaba fuera de los objetivos del proyecto.

El método más obvio para representar un grafo en una computadora es como una lista de átomos (nodos) y una lista de enlaces (arcos). Esta es esencialmente la estructura de los archivos MOLFILE (Dalby, 1992), como el que se puede ver en la Figura 4-1 para la molécula de ciclopropanona. Nótese que el formato MOLFILE incluye las coordenadas de los átomos, pero en este caso tienen un significado meramente estético. En programas más sofisticados se pueden usar estos campos de la tabla para almacenar verdaderas coordenadas atómicas tridimensionales.

La representación de las estructuras como tablas de átomos y enlaces ha sido la más usada en programas de CAOS, como OCSS y LHASA. En OSET usamos una tabla de enlace muy parecida al formato MOLFILE; de hecho, este es el formato de archivo que utiliza el programa. Usamos el formato MOLFILE porque es el formato más aceptado universalmente; casi todos los programas que trabajan con estructuras lo manejan.

Las matrices de enlace son matrices cuadradas de $n \times n$, donde n es el número de átomos en la molécula. El elemento m_{ij} de la matriz es igual al orden del enlace entre los átomos i y j . La diagonal de la matriz puede utilizarse para alguna propiedad de los átomos; por ejemplo, en el modelo de Dugundji-Ugi de matrices de enlace-electrón (*bond-electron matrix*), el elemento diagonal m_{ii} es igual al número de electrones de valencia no compartidos del átomo i . Además dado que la matriz sería simétrica ($m_{ij} = m_{ji}$, dado que los enlaces son bilaterales) es posible utilizar la mitad de estos elementos para representar alguna otra propiedad de los enlaces. En la Figura 4-1 se muestra la matriz de enlace-electrón para la ciclopropanona.

La ventaja principal de las matrices de enlace es que las operaciones aplicadas sobre las moléculas (transformadas y reacciones) son representadas de una manera simple como operaciones matriciales. Por ejemplo, para aplicar una reacción, se le suma a la matriz de enlace de la molécula una matriz de reacción; para el proceso inverso, es decir, la transformada, se resta la misma matriz de reacción. Como se mencionó en la página 17, todas las matrices de reacción pueden reducirse a un conjunto relativamente pequeño.

Finalmente, las notaciones lineales son muy usadas para almacenamiento de estructuras ya que son bastante compactas. Una notación lineal consiste en un conjunto de reglas que permiten transformar cualquier estructura en una cadena de caracteres. Por ejemplo, la fórmula semidesarrollada CH_3CH_3 para referirse al etano sería una notación lineal. El problema principal con las fórmulas semidesarrolladas es que no permiten describir fácilmente estructuras cíclicas. Las notaciones lineales tienen bases topológicas por lo que, como mencionamos anteriormente, no suelen tomar en consideración la estereoquímica. Sin embargo algunas de ellas han sido adaptadas para tomarla en cuenta.

Una de las más conocidas históricamente es la notación lineal de Wiswesser (WLN por su nombre en inglés), la cual se usó mucho en los años 70. La WLN tiene una serie de reglas relativamente complicadas (Vollmer, 1983) para generar una notación única² para cada estructura. La desventaja principal de la WLN es que tiene un gran número de reglas jerarquizadas, por lo que su aplicación puede ser complicada. Además, su lectura no es obvia porque usa diversas letras para representar subestructuras comunes. Su ventaja es que

es bastante compacta. Por ejemplo, la estructura de la ciclopropanona sería representada como L3VTJ en WLN. La lógica es la siguiente: L3 indica un anillo de tres átomos de carbono; V indica un grupo carbonilo; T indica que el anillo está saturado; y J se utiliza para marcar el final del anillo. Si hubiera cadenas sustituyentes aparecerían después de la J.

La notación lineal más usada en la actualidad es probablemente SMILES (*Simplified Molecular Input Line Entry System*; ver Bone, 1999). La ventaja principal de SMILES es que tiene un número muy pequeño de reglas simples y su lectura es bastante fácil ya que tiene cierto parecido con la notación semidesarrollada (Weininger, 1988). En general, la notación SMILES no es única, pero existe un algoritmo para obtener una notación única (Weininger, 1989). La desventaja de este algoritmo es que no es aplicable “a mano” pero, si consideramos que el uso de esta notación lineal siempre ha estado relacionado con aplicaciones computacionales, no resulta un problema importante. Como ejemplo, la estructura de la ciclopropanona se representa en SMILES como O=C1CC1. Todos los hidrógenos son implícitos; los dígitos sirven para señalar el cerrado del anillo.

A lo largo de las últimas tres décadas han aparecido varias propuestas de notaciones lineales (Ghoshal, 1990), pero aparentemente las únicas que han tenido éxito son las ya mencionadas. En el programa OSET utilizamos SMILES con fines de indizado y comparación debido a su simplicidad y a que es la notación lineal más utilizada en la actualidad, lo cual puede facilitar la integración con otros programas y bases de datos.

Percepción

Antes de analizar una molécula retrosintéticamente, es necesario percibir las características estructurales de interés químico: grupos funcionales, tipos de átomos, estereocentros, anillos, enlaces estratégicos, elementos de simetría, etc. Pare esto los programas de CAOS cuentan con un *módulo de percepción*, que se encuentra dividido a su vez en submódulos para la percepción de características específicas.

² Esto quiere decir que para cada estructura existe solamente una representación correcta en WLN. Esta propiedad es muy útil cuando se utiliza la notación con fines de indizado y de comparación.

Grupos funcionales

Los grupos funcionales son las subestructuras de una molécula que le confieren una reactividad característica, por lo que son los puntos de apoyo de la mayoría de las rutas sintéticas. La mayoría de los programas de síntesis necesitan identificar correctamente los grupos funcionales presentes en la molécula objetivo, para poder determinar las transformadas aplicables.

Al tratar de definir los grupos funcionales siempre surgen problemas que no resultan obvios al leer una definición como “el grupo funcional alcohol tiene una estructura ROH”. El problema está en la R . Por ejemplo, si la R es un carbono unido a un segundo oxígeno con un doble enlace, el grupo funcional ya no es un alcohol sino un ácido. Este caso es tan conocido que parece obvio, pero hay casos más ambiguos cuando se tienen dos grupos funcionales juntos. Por ejemplo, dos hidroxilos en un solo átomo presentan una reactividad muy distinta a la de dos hidroxilos en átomos separados, por lo que conviene considerarlos en la práctica como un nuevo grupo funcional, al que se le llama hidrato de carbonilo. Sin embargo, cuando se tienen dos halógenos en un solo átomo, no hay tanto problema: pueden considerarse para muchos fines como dos halógenos casi independientes.

Los enoles pueden considerarse desde cierto punto de vista como un híbrido entre un alcohol y un alqueno. Sin embargo, la reactividad de un enol es bastante diferente de la de los dos grupos funcionales originales, por lo que se prefiere considerarlo como un caso independiente. Estas son algunas de las dificultades a las que tienen que enfrentarse quienes trabajan en síntesis orgánica asistida por computadora, ya que a la computadora hay que programarla con reglas claras y objetivas.

La búsqueda de un grupo funcional en una molécula es equivalente al problema de encontrar un subgrafo dentro de un grafo. Este es un problema de programación clásico; se ha demostrado que tiene una gran complejidad computacional ya que crece de manera exponencial con el tamaño del grafo (Gondran, 1979), por lo que se han desarrollado diversos algoritmos con la intención de resolverlo de la manera más eficiente posible (Takeuchi, 1990). Los programas de CAOS suelen pasar una gran parte de su tiempo de ejecución detectando grupos funcionales (Bersohn, 1977). En OSET hemos encontrado que para algunas moléculas la identificación de grupos funcionales consume más del 70% del tiempo de procesamiento.

No todos los programas de CAOS identifican los grupos funcionales. Los programas con un fundamento más abstracto, como los de Hendrickson y Ugi, utilizan enfoques diferentes. Como se vio en el capítulo anterior, el SynGen de Hendrickson abstrae la situación de cada átomo esquelético de acuerdo con la electronegatividad de los átomos vecinos y los tipos de enlaces. Los programas de Ugi, son aun más abstractos y se concentran en la distribución formal de electrones en la molécula.

Anillos

La identificación de los anillos en la estructura es otra de las partes fundamentales de la percepción de la molécula objetivo. Hay tres razones para ello:

1. Es necesario identificar los anillos aromáticos de la molécula, ya que tienen una reactividad característica que se debe tomar en cuenta para el análisis retrosintético.
2. Existen muchas transformadas cuyo retrón incluye un anillo. Por ejemplo, la cicloadición de Diels–Alder, la anillación de Robinson, la adición de carbenos a dobles enlaces, la epoxidación, y diversas reacciones de síntesis de sistemas heterocíclicos.
3. Los anillos son el principal elemento de complejidad topológica, por lo que deben ser detectados para poder estimar la complejidad de la estructura, así como para la identificación de los enlaces estratégicos.

El problema de detección de anillos es otro problema clásico de la teoría de grafos y, al igual que el de búsqueda de subestructuras, es computacionalmente complejo. Para el caso de estructuras químicas se ha publicado una gran cantidad de artículos acerca de este problema; en un artículo de revisión de Downs (1989) se mencionan 24 algoritmos diferentes. Después de esta revisión han aparecido nuevos artículos; el más reciente parece ser el de Figueras (1996)

Canonización

En general hay más de una manera correcta de representar internamente una estructura mediante una tabla o matriz de enlaces. Para una molécula con n átomos no equivalentes, existen $n!$ (n factorial) permutaciones o maneras diferentes de numerar internamente los átomos. Aunque desde el punto de vista químico la numeración de los átomos es irrelevante, existen buenas razones para encontrar una manera única de numerar cada

estructura. Por ejemplo, la comparación de dos estructuras es uno de los problemas complejos de la teoría de grafos, pero si se tienen numerados los nodos de antemano de manera única se convierte en un problema trivial. La canonización resuelve este problema transformando la representación de la molécula en una *representación canónica*. El único requisito que tienen las representaciones canónicas es que para una estructura dada debe haber una y solamente una representación canónica válida.

Para dar un ejemplo que resulte familiar al químico, este es el mismo problema que se intenta resolver con las nomenclaturas sistemáticas como la de la IUPAC. Se desea que cada molécula tenga un solo nombre válido para poder indizarlas y buscarlas en bases de datos y catálogos como el *Chemical Abstracts*. Para ello se emplea un conjunto de reglas convencionales y en cierta medida arbitrarias. Y una gran parte de las reglas de la IUPAC tratan precisamente el problema de la numeración canónica.

En el caso de las aplicaciones computacionales, se puede utilizar cualquier conjunto de reglas, por arbitrarias que estas parezcan, que permita la obtención de una estructura canónica a partir de cualquier estructura. El algoritmo más conocido, y en el cual se basan muchos otros, es el de Morgan (1965). En un artículo de Bersohn (1977) se mencionan y comparan diversos algoritmos de canonización. Uno especialmente interesante es el de Wipke (1974) ya que toma en cuenta la estereoquímica. Como ejemplo de un algoritmo que no está basado en el de Morgan, se puede mencionar el de Ugi (Brandt, 1977).

El algoritmo usado en OSET es el propuesto por Weininger (1989) que se utiliza para generar representaciones SMILES canónicas. En el Capítulo 5 se describe con detalle. Todavía existe cierto interés en encontrar algoritmos de canonización eficientes: el más reciente es de Ouyang (1999).

Hay dos ventajas de usar representaciones canónicas en los programas de síntesis orgánica asistida por computadora:

1. Se facilita la comparación de estructuras. Esto permite buscar los precursores en catálogos de compuestos disponibles. Además, es posible comparar un precursor recién generado con otros analizados previamente, con el fin de evitar un análisis redundante.
2. La mayoría de los algoritmos de canonización detectan también los elementos de simetría estructural (*átomos equivalentes*). La simetría estructural es importante para la aplicación de ciertas estrategias topológicas (Corey, 1989). Además, hay ciertas

transformadas que requieren elementos de simetría estructural. Como ejemplo puede mencionarse la reacción de un éster con un exceso de reactivo de Grignard. El producto es un alcohol terciario donde por lo menos dos de las tres ramas son equivalentes.

Representación de las transformadas

Lo que más define a un programa de CAOS es el enfoque que utiliza para representar las transformadas. Esto está muy relacionado con el problema de clasificación visto en el Capítulo 2. Existen dos maneras principales de representar las transformadas: la manera algorítmica y la manera abstracta.

Representación algorítmica

La transformada se describe como una serie de operaciones a realizar, es decir, como un algoritmo. Este es uno de los enfoques más usados, y ya vimos un ejemplo con el lenguaje CHMTRN usado por LHASA (Figura 3-1). Este es el mismo enfoque que decidimos seguir para el programa OSET, como veremos en el Capítulo 5. La descripción de una transformada mediante este sistema consiste en dos etapas, que pueden estar separadas explícitamente o pueden estar mezcladas.

La primera etapa consiste en determinar si la transformada es aplicable. Para ello se examina el entorno del retrón en busca de características estructurales que pudieran descalificar el uso de la transformada. Por ejemplo, la presencia de un grupo saliente en la posición β a un grupo carbonilo imposibilita el uso de reacciones de condensación en medio básico, ya que predominaría el producto de eliminación. La segunda etapa consiste en la ejecución de la transformada, y es una especie de “receta” o “mecanismo”.

Una ventaja del enfoque algorítmico es que intenta representar de alguna manera el razonamiento que sigue el químico. Esto es especialmente útil para programas educativos, ya que es posible analizar el “razonamiento” de la máquina para tratar de entenderlo. Probablemente por ello es que este fue el primer enfoque utilizado, y es el que usamos en el programa OSET.

La base de datos de transformadas puede formar parte del programa principal y estar escrita en cualquier lenguaje de programación convencional como C, FORTRAN, BASIC, etc., o puede estar fuera del programa y escrita en un lenguaje especial. La primera opción

tiene la ventaja de que los lenguajes compilados son más rápidos y flexibles, pero resulta complicado modificar las transformadas; además, los lenguajes generales no están especialmente adaptados para la química por lo que resulta engorroso escribir las transformadas en ellos. Por eso, casi siempre se ha preferido la segunda opción, cuya ventaja principal es la facilidad de edición de la base de datos y una escritura más rápida y sencilla. Las transformadas residen en un archivo de texto externo al programa, por lo que es innecesario recompilar el programa cada vez que se agrega una transformada.

Otra ventaja del enfoque algorítmico es que solamente utiliza reacciones conocidas y probadas, por lo que suele ofrecer resultados más tangibles que los enfoques abstractos. Sin embargo, esta es también su principal desventaja: al explorar solamente el territorio conocido, el programa queda limitado por los conocimientos de su base de datos. Y se ha encontrado que mantener actualizada y completa la base de datos de un programa grande es una tarea laboriosa y costosa (Gund, 1980; Ihlenfeldt, 1995).

Representación abstracta

Existen dos enfoques principales para la descripción abstracta de las reacciones. el de Hendrickson (1992) y el de Ugi (1993), mencionados en el capítulo 2. La representación de Hendrickson se basa en la abstracción de las moléculas como esqueletos, donde cada átomo se caracteriza por propiedades como el número de enlaces σ y π y la electronegatividad de los vecinos. Las reacciones se representan como una suma de semirreacciones construidas a partir de los cambios de estas cuatro propiedades elementales. Este sistema es propio del programa SynGen, descrito en el Capítulo 3

El sistema de Ugi se basa en la suma y resta de matrices de reacción a matrices de enlace-electrón. Aunque existe una cantidad enorme de matrices de reacción, éstas pueden ser reducidas a un conjunto de aproximadamente treinta matrices. Hay muchas maneras potenciales de aplicar una matriz de reacción reducida a una matriz de enlace-electrón, por lo que es necesario elegir únicamente combinaciones razonables. Para ello se usan la heurística y criterios termoquímicos de estabilidad.

4.2 Operaciones avanzadas

Las operaciones avanzadas no son necesarias para todos los programas de CAOS; se necesitan especialmente en los programas no interactivos, ya que éstos requieren un grado mayor de automatización. En los programas interactivos más simples, las operaciones avanzadas recaen completamente en el usuario.

Tácticas

Las tácticas en síntesis orgánica tienen la función de alcanzar eficientemente la transformación deseada (Smit, 1998). Puede decirse que las tácticas sirven para resolver los problemas específicos y de corto plazo en una síntesis. Hay diversos tipos de tácticas, entre las que se encuentran la protección/desprotección, activación, intercambio de grupos funcionales, uso de grupos funcionales enmascarados, control cinético/termodinámico, etc.

Aunque es posible y recomendable planear las tácticas con antelación, muchas veces hay tácticas que son improvisadas en el laboratorio. Por eso son mejores los planes sintéticos flexibles, que permiten cambiar de táctica si el plan original falla.

Muchos programas dejan que el químico se encargue de las tácticas, pero hay algunos que por lo menos ayudan. Programas como CAMEO pueden ser útiles en el análisis de pasos específicos de la síntesis con el fin de predecir posibles subproductos. En los programas como LHASA, las tácticas dependen del grado de detalle con el que se encuentren descritas las transformadas en la base de datos. Por ejemplo, LHASA puede identificar los posibles problemas de selectividad, ya que tiene una tabla de sensibilidad de los grupos funcionales ante más de un centenar de condiciones de reacción típicas (Stolow, 1980). El programa PROTECT, que forma parte del proyecto LHASA (Corey, 1985b), sirve para ayudar en la selección de grupos protectores para una síntesis.

Hay que tener cuidado al considerar las condiciones de reacción "típicas", ya que puede haber muchas variaciones. Según Smit (1998, p. 103) existen más de cuarenta métodos diferentes para lograr la transformación $R-OH \rightarrow R-Cl$. Pero, como todo buen químico sabe, la elección de un método u otro puede depender más del inventario del laboratorio que del plan de síntesis.

Estrategias

Las estrategias están relacionadas con el plan global de la síntesis, es decir, con el árbol retrosintético y la molécula como un todo (Smit, 1998). Es necesario decidir en qué orden se van a construir las diversas partes de la molécula, cuáles son los enlaces considerados “estratégicos”, etc. En el Capítulo 2 ya mencioné algunos de los principios estratégicos principales que ayudan en el diseño de rutas sintéticas.

El programa que más ha desarrollado el uso de estrategias es LHASA, como se vio en el Capítulo 3. LHASA requiere que el usuario seleccione la estrategia que desea utilizar antes de comenzar el análisis. Sin embargo, hay otros programas donde alguna estrategia es parte integral del diseño del programa. Por ejemplo, SynGen está diseñado para buscar siempre síntesis convergentes y con un mínimo de refuncionalizaciones. Esta fue una decisión estratégica por parte del autor del programa y no requiere decisiones por parte del usuario.

Cuando se analiza retrosintéticamente el árbol de síntesis, la principal función de las estrategias es el *podado* del árbol. Barone (1986) identificó cinco tipos de podado:

1. **Podado por límite de profundidad.** Este tipo se utiliza en los programas no interactivos para evitar que la búsqueda se pierda en una rama demasiado profunda. El usuario puede especificar que quiere que la búsqueda tenga un máximo de n pasos; de esta manera se puede controlar el tiempo de ejecución del programa y la longitud máxima de la síntesis propuesta (Takahashi, 1990).
2. **Búsqueda de ramas con formas específicas.** Esto generalmente consiste en la selección de rutas convergentes. Puede demostrarse fácilmente que, para el mismo número de pasos con rendimiento similar, las síntesis convergentes siempre producen un rendimiento global mayor. Además, tienen la ventaja de que pueden ser paralelizadas y generalmente son más flexibles. Este es el enfoque usado por SynGen (Hendrickson, 1990).
3. **Podado temprano.** Para ello se utilizan criterios heurísticos como las estrategias de Corey, y especialmente las estrategias topológicas. Una vez identificados los enlaces estratégicos, se podan las ramas que no conduzcan a la desconexión de alguno de ellos. Este tipo de podado tiene la gran ventaja de que reduce sensiblemente el tiempo de cómputo, pero presenta el riesgo de eliminar excelentes rutas sintéticas que no

parecerían prometedoras al principio del análisis. Nunca hay que olvidar que las reglas heurísticas son falibles. Haciendo una analogía con el ajedrez: todo el mundo sabe que no es bueno permitir que el oponente capture su reina (regla heurística); sin embargo, hay ocasiones en las que el sacrificio de la reina puede ser el mejor camino y conducir a la victoria (Seirawan, 1995).

4. **Podado de ramas que no van en cierta dirección.** En programas que hacen búsquedas de largo alcance con estrategias basadas en transformadas o en materias primas, se eliminan las ramas que no aumenten la semejanza con la materia prima o el retrón deseado. LHASA es un buen ejemplo de este tipo de programa.
5. **Podado difuso.** Se detiene el análisis de una rama cuando se cumple cierta característica estructural especificada. Por ejemplo, la presencia del precursor en un catálogo (Gelernter, 1977), o el hallazgo de un precursor suficientemente simple, medido con alguna función de complejidad arbitraria (Takahashi, 1990). Las funciones de complejidad toman en cuenta factores como el número y tipo de átomos, grupos funcionales y anillos.

5. FUNCIONAMIENTO DEL PROGRAMA OSET

OSET es un sistema cliente-servidor. Esto quiere decir que el trabajo está distribuido entre dos computadoras conectadas en red. En cada una corren programas diferentes con funciones específicas:

- **Cliente.** Se encarga de la interfase con el usuario. Esto incluye la entrada y salida de datos, dibujo de estructuras, y navegación del árbol retrosintético.
- **Servidor.** Se encarga del análisis de las moléculas.

¿Por qué elegimos un enfoque cliente-servidor, en lugar de hacer un programa sencillo que corra en una sola máquina?

El sistema cliente-servidor que utilizamos tiene las siguientes ventajas:

- En nuestro caso, el cliente es un *applet Java*, por lo que puede ejecutarse en cualquier computadora que tenga instalado un *browser* razonablemente reciente, como el Internet Explorer o Netscape versiones 4.0 o superiores.¹ Con esto logramos que el programa sea utilizable con cualquiera de los sistemas operativos más usados en la actualidad: Windows, MacOS y Linux.
- El cliente es un programa pequeño (menos de 60 kilobytes), y por ser un applet no necesita ser instalado sino solamente bajado del servidor cada vez que hay una nueva versión (esta operación toma alrededor de 15 segundos en una conexión por módem de 56 kbps). Pero como el cliente es sencillo y tiene funciones bien definidas, una vez que está terminado su desarrollo casi no sufre modificaciones.
- La base de conocimiento del sistema (base de datos de reacciones) está en el servidor. Aunque los datos son actualizados constantemente, ya no es necesario que el usuario

¹ Al momento de redactar esta tesis, las versiones más recientes de estos programas eran Internet Explorer 5.5 y Netscape 6.0

actualice constantemente su versión del programa. Incluso sin darse cuenta, estará usando siempre la versión más reciente de la base de datos.

- El servidor, como no tiene interfase directa con el usuario y está escrito en un lenguaje estándar (ANSI C), es un programa portátil, lo que quiere decir que se puede compilar y ejecutar en diversos sistemas operativos. Nosotros lo hemos probado exitosamente en Windows y en Linux.
- El programa servidor puede instalarse en una máquina poderosa.
- Como efecto colateral, ya que hemos diseñado el programa para funcionar en red, sería relativamente fácil hacer que corra de manera distribuida entre varias computadoras, acortando así el tiempo de análisis.

La principal desventaja es que para usar el programa se necesita una computadora conectada a la red. Pero actualmente la gran mayoría de las computadoras de uso académico están conectadas a la red, así que no es tan grande ese problema. Como servidor, el usuario puede usar el servidor que tenemos instalado en <http://litio.pquim.unam.mx/oset/> o instalar el suyo propio. Cabe mencionar que es posible usar la misma computadora simultáneamente como cliente y como servidor, con lo cual el sistema funcionaría de manera similar a un programa convencional.

En el resto de este capítulo describiré con mayor detalle el funcionamiento interno del cliente y del servidor. En cada una de las secciones se incluye una *lista de funciones relacionadas*. Estas son las partes del programa que realizan las operaciones de la sección correspondiente y se incluyen como guía por si el lector está interesado en revisar el código fuente. Se encuentran en el formato archivo: `función()`.

5.1 Esquema general

En la Figura 5-1 se puede ver de manera general la relación y las interconexiones que existen entre el cliente y el servidor. En las secciones que siguen veremos con detalle en qué consiste y para qué sirve cada una de las operaciones que aparecen en el esquema. El uso del sistema puede resumirse de la siguiente manera:

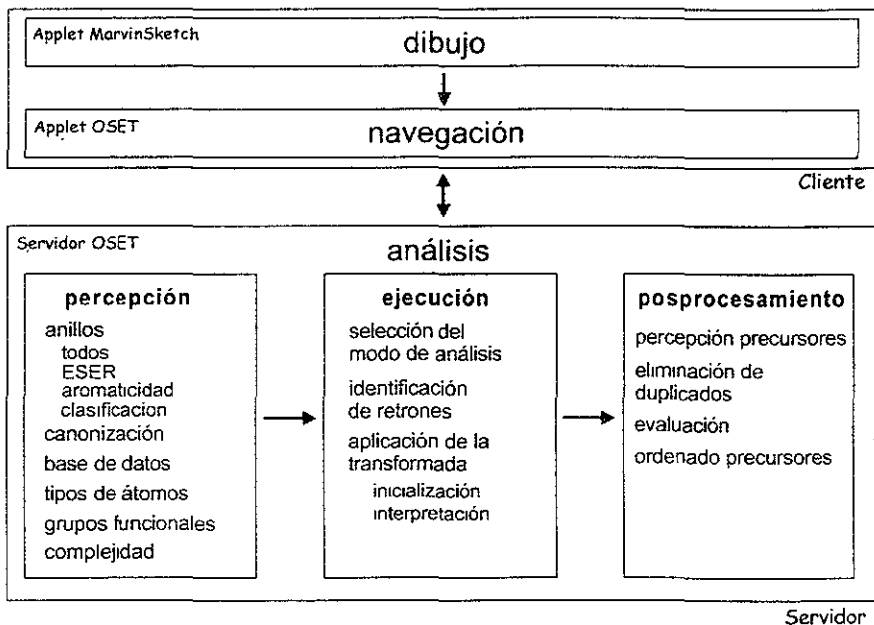


Figura 5-1 Esquema general del funcionamiento del sistema OSET.

1. El usuario dibuja la molécula objetivo en el applet MarvinSketch, y presiona el botón *analyze*.
2. Usando el applet OSET, el usuario navega la parte ya analizada del árbol (que al principio incluye solamente la molécula objetivo) y escoge una molécula para analizar.
3. El applet envía la estructura al servidor, el cual la analiza y devuelve los resultados al cliente. El applet muestra los resultados, y se regresa al paso 2. Por eso la flecha que une al cliente con el servidor en el esquema es doble.

5.2 Cliente

Dibujo de la estructura

Como puede verse en la Figura 5-1, el cliente está dividido en dos componentes. Para la parte de dibujo de estructuras, utilizamos un applet disponible gratuitamente llamado

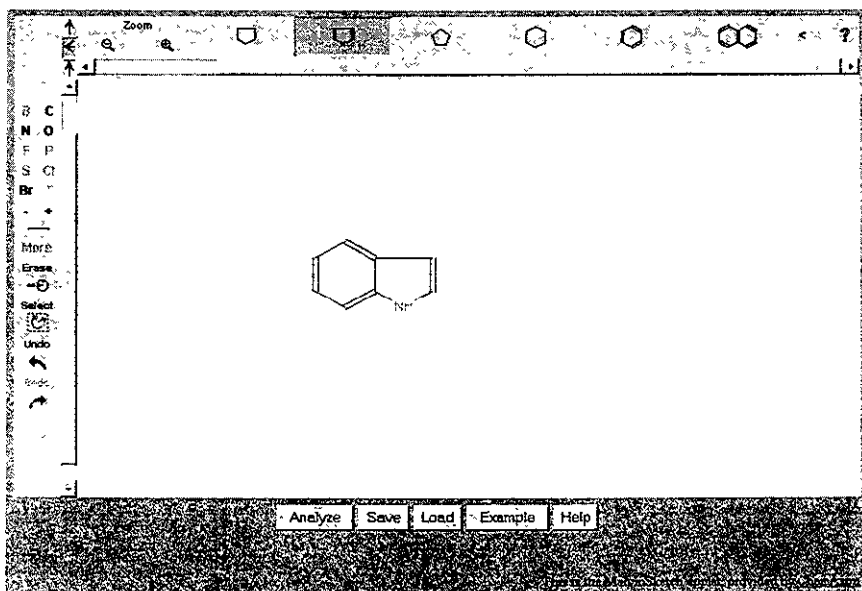


Figura 5-2 Applet MarvinSketch.

MarvinSketch, elaborado por ChemAxon (Csizmadia, 2000).² En la Figura 5-2 se puede ver una muestra de la pantalla del applet. El usuario dibuja o importa una estructura utilizando dicho applet, y al presionar el botón *analyze* se invoca el applet OSET, que es el que se usa para realizar el análisis retrosintético en sí.

Navegación del árbol

Una vez que el usuario presiona *analyze* se entra al applet de OSET propiamente dicho. Su función es esencialmente la navegación del árbol retrosintético. La pantalla del applet OSET se muestra en la Figura 5-3. En el Apéndice A se explica con detalle el uso y funcionamiento de cada uno de los elementos de la interfase gráfica.

Antes de analizar una molécula, el usuario debe especificar el modo de análisis deseado. En la página 66 se detallan los cuatro modos de análisis que maneja el servidor, pero el cliente solamente maneja tres: desconexión, auxiliar y desconexión sin retrón. El

² Disponible en <http://www.chemaxon.com/>

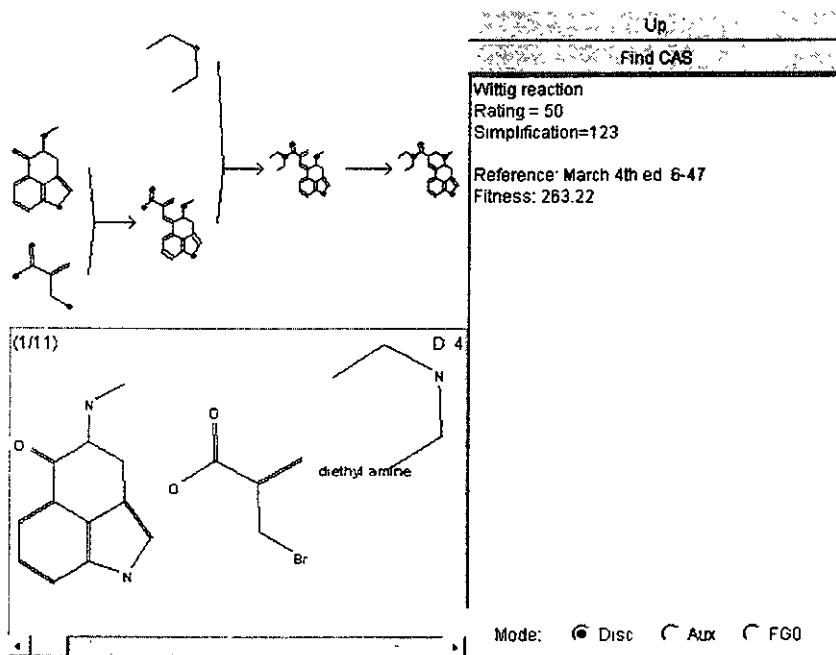


Figura 5-3 Applet OSET.

servidor considera las transformadas tipo FGI y FGA como dos modos de análisis diferentes, pero para simplificar la interfase las agrupamos en el cliente bajo el nombre genérico de “auxiliar”.

El cliente guarda en memoria la parte del árbol que ya ha sido explorada, de tal suerte que si el usuario intenta analizar una molécula que ya analizó anteriormente, el applet puede mostrar los resultados rápidamente sin tener que comunicarse con el servidor.

Cuando el usuario selecciona una molécula nueva para analizar, el cliente abre una conexión mediante el protocolo TCP/IP³ hacia la computadora donde se ejecuta el programa servidor (que puede ser la misma donde se ejecuta el cliente). Utilizando un protocolo especialmente creado al que hemos denominado *Protocolo CAOS*, el cliente envía al servidor toda la información necesaria sobre la molécula y solicita el modo de

³ *Transfer Control Protocol / Internet Protocol* Es el protocolo de comunicaciones utilizado generalmente en internet. Considero que queda fuera del alcance de esta tesis entrar en detalles acerca de los protocolos.

análisis elegido. El servidor contesta entonces con una lista de precursores. El cliente los agrega al árbol y procede a mostrarlos.

El servidor no tiene noción alguna del árbol retrosintético: analiza una molécula a la vez y se olvida del asunto hasta que recibe una nueva molécula para analizar;⁴ esto le facilita la interacción simultánea con varios clientes. El cliente es el encargado de manejar el árbol completo, por lo que es la parte que puede evaluar globalmente una ruta sintética. Con este fin hemos probado una función a la que hemos llamado *fitness* (aptitud). La idea es que las buenas rutas sintéticas minimicen esta función. Si fuera posible encontrar una función que realmente lograra ese propósito, sería factible hacer que la computadora explore el árbol automáticamente e identifique las mejores síntesis. Esta es una línea de desarrollo futuro sumamente interesante, pero se sale del propósito educativo enunciado en esta tesis, ya que la idea es que sea el estudiante quien explore el árbol y evalúe las rutas sintéticas.

5.3 Servidor

Percepción → ejecución → posprocesamiento

Como puede verse en el esquema general, la función principal del servidor es analizar estructuras de moléculas. El análisis consta de tres etapas: percepción, ejecución y posprocesamiento.

Percepción

Anillos → canonización → búsqueda en base de datos →
clasificación de átomos → grupos funcionales → complejidad

En el capítulo anterior mencioné la función de la percepción en general y de algunas de sus etapas principales. En el esquema se puede ver el orden en el que se realizan los pasos en el programa OSET. El orden en el que se realizan los pasos no es el único posible, pero tampoco es totalmente arbitrario porque algunos pasos del proceso de percepción dependen

⁴ En términos computacionales se dice que es un servidor sin estado (*stateless*)

de otros. Por ejemplo, la clasificación de átomos depende de la clasificación de anillos; y el cálculo de la complejidad depende de todos los demás pasos.

Funciones relacionadas

```
_analyze.c: parse_mol()
```

Anillos

Todos → ESER → aromaticidad → tipos

La búsqueda de los anillos es una parte muy importante de la percepción. Permite identificar anillos aromáticos, sistemas heterocíclicos, y calcular la complejidad de la molécula tomando en cuenta sus anillos. La búsqueda de anillos está a su vez dividida en varias etapas.

Funciones relacionadas

```
__molring.c: find_rings()
```

Todos los anillos

Usamos el algoritmo propuesto por Hanser (1996). Decidimos usar un algoritmo que encontrara todos los anillos, y no solamente una base mínima como hacen la mayoría de los algoritmos, porque en muchas estructuras hay anillos que no pertenecen a la base mínima y sin embargo son de interés sintético (por ejemplo, el anillo de seis átomos en el norbornano).

La desventaja de la búsqueda exhaustiva de anillos es que los sistemas policíclicos grandes pueden tener millones de anillos. De hecho, el número máximo posible de anillos en una estructura crece exponencialmente como $2^n - 1$, donde n es el número de anillos en la base mínima. En una ocasión intentamos probar este algoritmo con la molécula de buckminsterfullereno (C_{60}). No terminó de ejecutar en un tiempo razonable. Desconozco el número de anillos presentes en la estructura de C_{60} , pero el número máximo teórico es $2^{31} - 1 = 2,147,483,647$.

Funciones relacionadas

```
_molring.c: findallrings()
```

Conjunto esencial de anillos esenciales (ESER, *Essential Set of Essential Rings*)

Una vez que se tienen todos los anillos, es necesario seleccionar el subconjunto de anillos que pueden tener interés sintético. Para ello usamos el algoritmo propuesto por Fujita (1988), que genera el llamado *conjunto esencial de anillos esenciales*. Este algoritmo toma en cuenta el tipo de relación entre los anillos (fusiones, puentes, etc.) y la presencia o ausencia de heteroátomos en los mismos, con la intención de generar un subconjunto intuitivamente “esencial”.

Este algoritmo ha mostrado tener ventajas y desventajas. Por ejemplo, en el cubano se obtiene un ESER que consta de los seis anillos de cuatro átomos, es decir las seis caras del cubo, mientras que los algoritmos que generan bases mínimas encontrarían solamente cinco de las caras del cubo, elegidas de manera indeterminada. Sin embargo, en sistemas como las porfirinas el algoritmo de Fujita ha encontrado una gran cantidad de anillos de dudosa utilidad, mientras que los algoritmos que buscan bases mínimas encontrarían solamente cinco anillos (una cantidad más razonable).

Funciones relacionadas

```
_molring.c: find_rings()
```

Detección de aromaticidad

Esta etapa es esencial para la posterior aplicación de transformadas retrosintéticas correspondientes a reacciones como la sustitución electrofílica aromática, la sustitución nucleofílica aromática, y síntesis de heterociclos aromáticos. La detección de aromaticidad es necesaria debido a que, internamente y en diversos formatos de archivo, las moléculas aromáticas son representadas mediante enlaces sencillos y dobles alternados.

El algoritmo está basado en la regla de Hückel: en principio se consideran como aromáticos los anillos planos, que contengan $4n + 2$ electrones pi y donde todos los átomos tengan hibridación sp^2 . En la práctica simplificamos un poco la regla y solamente la aplicamos a anillos de cinco a siete átomos, ya que en el caso de sistemas más grandes es

difícil constatar la planaridad del sistema sin información adicional, y la gran mayoría de los sistemas aromáticos de interés son anillos de cinco y seis miembros.

El conteo de electrones pi no es trivial, ya que debe tomar en cuenta las diferentes estructuras resonantes, donde no siempre se encuentran los dobles enlaces en el anillo que se está analizando. En la forma resonante **1** del naftaleno (Figura 5-4) es fácil identificar dos anillos bencénicos, mientras que en la forma **2** el anillo de la izquierda es evidentemente un benceno, mientras que el anillo de la derecha tiene solamente dos enlaces dobles endocíclicos.



Figura 5-4 Formas resonantes del naftaleno

Para remediar este problema, decidimos tomar en cuenta también los electrones pi de los enlaces exocíclicos con átomos no más electronegativos que el carbono, siguiendo la propuesta de Weininger (1988). De esta manera, compuestos como las quinonas son correctamente considerados como no aromáticos, pero compuestos como el [6]radialeno (Hopf, 1992) son erróneamente considerados como aromáticos (ver Figura 5-5) Es difícil establecer un conjunto de reglas infalibles, pero considero que el adoptado funciona bien en la mayor parte de los casos.⁵

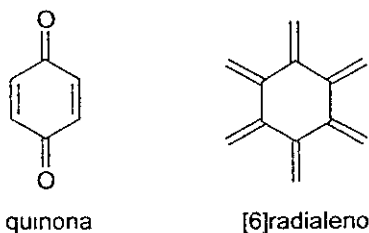


Figura 5-5 Ejemplos de compuestos no aromáticos con enlaces exocíclicos. Actualmente nuestro algoritmo clasifica erróneamente como aromático al [6]radialeno.

⁵ No olvidemos que el radialeno y algunas de las otras moléculas problemáticas son altamente inestables por lo que se les considera casos poco comunes.

Por supuesto, también hay que considerar los compuestos heterocíclicos aromáticos. Por ahora decidimos limitarnos a heterociclos con nitrógeno, oxígeno y azufre. La regla es sencilla: los átomos de estos elementos que no participen en dobles enlaces contribuyen con dos electrones pi, excepto los nitrógenos cargados positivamente que tienen hibridación sp^3 y no contribuyen. Los átomos que participan en un doble enlace contribuyen con un electrón pi, independientemente de que sean heteroátomos o no.

También hay que tomar en cuenta la posibilidad de átomos de carbono cargados, como los que ocurren en el anión ciclopentadienilo y el catión tropilio (Figura 5-6). Los carbonos con carga negativa contribuyen con dos electrones pi, mientras que los átomos de carbono cargados positivamente contribuyen con un orbital p vacío.

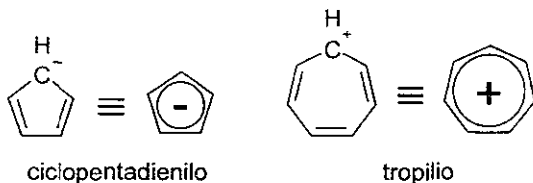


Figura 5-6 Aromaticidad de carbaniones y carbocationes orgánicos. En ambos casos se tienen seis electrones pi.

La razón por la que la detección de aromaticidad debe realizarse antes de la canonización es que en el formato SMILES, que utilizamos internamente como representación canónica, se distinguen los átomos aromáticos de los alifáticos.

Funciones relacionadas

`_aroma.c: detect_aromaticity()`

Clasificación de anillos

Una vez que se han detectado los anillos esenciales y se ha determinado la aromaticidad de cada uno de ellos, se procede a identificar y clasificar los tipos específicos de anillos de interés sintético. Para ello se utiliza una lista ubicada en el archivo `rings.chm`. En este archivo se describen anillos tales como el benceno, piridina, pirrol, furano, ciclohexeno, ciclopropano, etc., todos estos tipos de anillos tienen transformadas retrosintéticas particulares. También se incluyen sistemas fusionados como el indol y la quinolina. Como

ejemplos de la manera en que se describen los anillos veamos la piridina, el ciclohexeno y el indol:

```
pyridine = "nccccc", 25
cyclohexene = "C=CCCC", 10
indole = (pyrrole, A=[5,4]) (benzene, A=[2,3]), 40
```

La descripción consta de un nombre, una “fórmula” y un valor de complejidad. Para los anillos simples, la fórmula consiste en una lista de átomos, con los enlaces múltiples especificados si es necesario (como en el caso del ciclohexeno). Los anillos aromáticos se distinguen escribiendo todos los símbolos atómicos con letras minúsculas (como en la piridina). Los anillos fusionados se describen como la unión de dos o más anillos simples; para cada anillo se especifica un enlace de fusión, el cual consiste en una letra para identificarlo y un par de átomos para definirlo. En el indol decidimos identificar el enlace de fusión con la letra A y utilizamos el enlace entre los átomos 5 y 4 del pirrol, fusionado con el enlace entre los átomos 2 y 3 del benceno.

Funciones relacionadas

```
_molring.c: classify_rings()
```

Canonización

Usamos directamente el algoritmo propuesto por Weininger (1989) para numerar los átomos de la estructura de manera canónica y detectar los elementos de simetría. Una vez numerados los átomos, se genera un SMILES canónico, el cual es una cadena de caracteres y se utiliza principalmente para comparar estructuras. Esto sirve en el posprocesamiento para eliminar resultados duplicados y para buscar las estructuras en bases de datos.

El algoritmo de numeración canónica consiste básicamente en lo siguiente (para una explicación más detallada, consultar la referencia):

1. Se asigna a cada átomo una *función invariante* que depende del número atómico, número y tipo de enlaces, carga, y número de átomos de hidrógeno vecinos
2. Se ordenan los átomos según el valor de la función invariante y a cada uno se le asigna secuencialmente un *número de clase*. Los átomos con igual valor de la función invariante tienen el mismo número de clase.

3. A cada átomo se le asigna un número primo que, para un átomo clase i , es el elemento i -ésimo de la secuencia de los números primos (2, 3, 5, 7, 11, ...).
4. A cada átomo se le asigna un valor de ordenamiento que es igual al producto de los números primos de todos sus vecinos inmediatos.
5. Se ordenan los átomos según este producto y se asignan nuevos números de clase. Los átomos con igual producto de números primos siguen teniendo igual número de clase.
6. Si aumentó el número de clases diferentes, regresar al paso 3. Si no aumentó, continuar en el paso 7.
7. En la primera pasada, registrar las clases de simetría. Los átomos con igual número de clase son estructuralmente equivalentes.
8. Si el número de clases diferentes es igual al número de átomos, FIN.
9. Multiplicar todos los números de clase por 2 y restarle 1 a cualquiera de los átomos de la primera clase degenerada.
10. Ir al paso 3.

Funciones relacionadas:

`_canon.c: canonicalize()`

`_smiles.c: mol2smiles()`

Búsqueda en la base de datos de materias primas

Una vez que se ha canonizado la estructura, se busca en la base de datos de materias primas. Esta base de datos es un archivo de texto llamado `sm.chm` y contiene el nombre de cada materia prima con su correspondiente SMILES canónico. Actualmente el archivo contiene alrededor de ochenta compuestos, pero puede ser extendido fácilmente. Los compuestos seleccionados son las que consideramos materias primas "de libro de texto"; generalmente se trata de compuestos alifáticos con menos de cinco átomos de carbono y un grupo funcional o compuestos aromáticos simples.

Esta etapa tiene dos propósitos: primero, sugerirle al usuario que ya encontró una materia prima ampliamente disponible y que no necesita continuar el análisis en esa dirección; segundo, a las materias primas que aparecen en la lista se les asigna una complejidad menor en la etapa de cálculo de complejidad.

methyl	primary	secondary
tertiary	quaternary	vinyl
carbonyl	alkynyl	nitrile
allene	alkyl	sp3
sp2	sp	ether
nitrile	hydroxyl	nitro
peroxide	carbonyl	benzyl
allyl	alpha_nitrile	alpha_carbonyl
alpha_alkynyl	alpha_CH	phenyl
EWG	alpha_EWG	alpha_EWG2
alpha_EWG3	C	N
H	O	S
X	Cl	Br
I		

Figura 5-7 Tipos de átomos que reconoce el programa. Los tipos no son excluyentes, es decir, un átomo dado puede pertenecer a varias categorías al mismo tiempo.

Funciones relacionadas:

```
_smdb.c: findsmiledb()
```

Clasificación de átomos

Muchas reacciones tienen una aplicación limitada por el tipo de átomos involucrados. Por ejemplo, aquellas que tienen un mecanismo S_N2 difícilmente funcionarán si el sustrato es terciario. Por eso es necesario clasificar todos los átomos de la molécula, y especialmente los de carbono. Los heteroátomos generalmente pertenecen a un grupo funcional, lo cual es suficiente en la mayoría de los casos para conocer su entorno. Pero los átomos de carbono muchas veces están alejados de todo grupo funcional, por lo que se torna indispensable un análisis adicional de sus alrededores.

Los átomos se clasifican en tipos generalmente no excluyentes, lo que quiere decir que un átomo puede ser de varios tipos simultáneamente. Por ejemplo, un átomo de carbono podría clasificarse como secundario, alquílico y alílico a la vez. Actualmente tomamos en cuenta cerca de cuarenta tipos de átomos (Figura 5-7)

Los patrones usados para la clasificación de átomos se encuentran en el programa principal. Esto tiene la ventaja de que la clasificación es más rápida que si se usaran patrones externos tomados de un archivo (como en el caso de los grupos funcionales que veremos más adelante), pero tiene la desventaja de que no es posible agregar nuevos tipos de átomos sin alterar el código del programa y recompilarlo. Tomamos esta decisión considerando que, dado que la clasificación de átomos hay que realizarla siempre para todos los átomos de una molécula, era más importante la velocidad que la flexibilidad.

Además, consideramos que los tipos de átomos son más fáciles de definir de manera suficientemente exhaustiva que los grupos funcionales.

La razón por la que la clasificación de átomos se realiza después de la identificación de anillos es que depende de la detección de aromaticidad. Por ejemplo, existen los átomos de tipo bencílico y fenílico, cuya identificación depende evidentemente de la aromaticidad.

Funciones relacionadas:

```
_clas2.c: classify_atoms()
```

Identificación de grupos funcionales

La mayoría de las transformadas retrosintéticas están relacionadas con grupos funcionales. Por ello es eminentemente necesario identificar los grupos funcionales presentes en la molécula. El algoritmo del programa OSET permite la definición de cualquier cantidad de grupos funcionales, los cuales se encuentran en el archivo `funcgrp.chm`. Los grupos funcionales se definen como un árbol (en el sentido topológico). A partir de un *átomo raíz* se definen los demás átomos del grupo funcional como las ramas y hojas del árbol (Figura 5-8). El átomo raíz puede escogerse arbitrariamente, pero es recomendable usar una notación consistente ya que a partir de este átomo se numeran los demás para su utilización en la descripción de las transformadas.

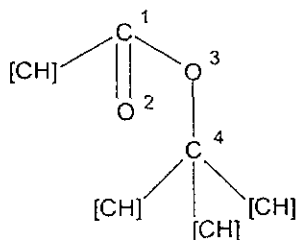


Figura 5-8 Descripción de un grupo funcional como estructura de árbol. En este ejemplo se muestra el grupo funcional éster. Los elementos entre corchetes representan opciones, en este caso, carbono o hidrógeno.

phenol	alcohol	enol_ether
amine1	amine2	amine3
acid	olefin	ketone
ester	aldehyde	cyanide
amide	ether	alkyne
alkyl_chloride	aryl_chloride	acyl_chloride
alkyl_bromide	aryl_bromide	vinyl_bromide
acetal	nitro	thiol
alkyl_iodide		

Figura 5-9 Grupos funcionales reconocidos por OSET

La definición de un grupo funcional en el archivo `funcgrp.chm` tiene la siguiente forma:

```
ESTER = "C(=O O(C([CH]^3) [CH]))", 10
```

Al igual que en el archivo de definición de tipos de anillos, la definición consiste en un nombre, una fórmula y un valor de complejidad. En el Apéndice B se encuentra una descripción detallada de las reglas de descripción de grupos funcionales. Actualmente se encuentran definidos más de veinte grupos funcionales (Figura 5-9), pero es fácil agregar más si se desea.

Existe más de una manera de definir la mayoría de los grupos funcionales. En el caso del éster, por ejemplo, podríamos haber usado como raíz alguno de los átomos de oxígeno o el otro átomo de carbono. Sin embargo, hay que hacer hincapié en que el orden en el que aparecen los átomos en la definición del grupo *sí* es relevante, ya que se utiliza para la asignación de las variables predeterminadas de tipo átomo utilizadas en la descripción de cada transformada. Éstas se determinan numerando de izquierda a derecha los átomos en la definición del grupo funcional: en el éster, el carbono de carbonilo sería el átomo 1, el oxígeno de carbonilo el 2, y el otro oxígeno y el otro carbono serían los átomos 3 y 4 respectivamente.

Funciones relacionadas:

```
_fgfind.c: busca_funcionales()
```

Cálculo de complejidad

El objetivo del análisis retrosintético es simplificar la molécula objetivo. Para poder evaluar el grado de simplificación alcanzado mediante la aplicación de una transformada, es necesario comparar la complejidad de la molécula objetivo con la de los precursores. Para

ello se calcula para cada estructura una función empírica de complejidad. Esta función da por resultado un número entero, y depende de los siguientes factores:

- Número de átomos
- Grado de ramificación
- Número y tipo de anillos
- Número y tipo de grupos funcionales
- Carácter de materia prima

Para cada factor hay parámetros que hemos ajustado de manera empírica para que la función de complejidad dé los resultados que el químico esperaría de manera intuitiva. Algunos de ellos pueden ser modificados por el usuario en los archivos correspondientes.

Sobra decir que el cálculo de la función de complejidad debe ser el último paso de la percepción, ya que depende de casi todos los anteriores.

Funciones relacionadas:

`_complex.c: mol_complexity()`

Ejecución

Selección de modo de análisis → identificación de retronos → aplicación de transformada

Una vez realizada la percepción de la molécula, se procede a aplicar las transformadas retrosintéticas apropiadas. Las transformadas se encuentran en el archivo `rxns.chm` y los archivos a los que éste haga referencia. Como ejemplo, en la Figura 5-10 se muestra una de las transformadas de la base de datos de OSET. La explicación del lenguaje utilizado se encuentra en el Apéndice B.

Selección del modo de análisis

El servidor utiliza el modo de análisis escogido por el usuario en el cliente. Los cuatro modos de análisis disponibles son los siguientes:

- 1 **Desconexión.** El modo preferido normalmente: aplica solamente transformadas que desconectan enlaces C-C o esqueléticos

```

.RXN
NAME "Aldol condensation"
TYPE GP2
G1 KETONE, aldehyde, amide, ester
G2 ALCOHOL
PATH=3
RATING=50
link "reactions/aldol.html"

.CONDITIONS
    STRONG_BASE

.COMMENTS
    "Usually done under moderately basic conditions.\n"
    "Reference: March 4th ed. 6-39"

.START
if(b1 isnot tertiary) then
    echo("Addition to aldehyde: Rating increased")
    rating = rating+5
endif
rs = ringsize(p2, p3)
if(rs > 0) then
    if((rs <= 6) and (rs >= 5)) then
        rating = rating + 10
        echo("Intramolecular: rating increased")
        BREAKBOND(P2, P3)
        MAKEBOND(B1, B2)
        DONE
    endif
else
    BREAKBOND(P2, P3)
    MAKEBOND(B1, B2)
    DONE
endif
endif

```

Figura 5-10 Ejemplo de transformada en el lenguaje usado por OSET

- Intercambio de grupo funcional (FGI).** Uno de los llamados *modos auxiliares*, incluye las transformadas que cambian un grupo funcional por otro. Dada la gran cantidad de reacciones FGI disponibles, con el fin de mantener manejable la cantidad de resultados, al utilizar este modo de análisis es necesario especificar el grupo que se desea transformar.
- Adición de grupo funcional (FGA).** Este modo auxiliar agrega un grupo funcional donde antes no había ninguno. En este caso es necesario especificar el *átomo* donde se desea agregar el nuevo grupo funcional, ya que de lo contrario se obtendrían decenas de resultados inútiles.
- Desconexión sin retrón (GP0).** Esta categoría incluye las reacciones de acoplamiento que no dejan grupo funcional alguno en el producto. Por ello, no es posible identificar

Tipo de transformada	Información del retrón	Información adicional
Desconexión de un grupo funcional	Grupo funcional	
Desconexión de dos grupos funcionales	Dos grupos funcionales y la longitud del camino que los separa.	
Desconexión de anillo	Tipo de anillo	
Intercambio de un grupo funcional	Grupo funcional	Grupo funcional precursor
Intercambio de dos grupos funcionales	Dos grupos funcionales y la longitud del camino que los separa.	Grupo funcional precursor
Adición de grupo funcional	No hay. El usuario debe especificar un átomo	Grupo funcional precursor
Desconexión sin retrón	No hay. El usuario debe especificar un enlace	

Tabla 5-1 Información de retronos contenida en el encabezado de la transformada.

un retrón en la molécula objetivo que indique el uso de dicha transformada. Al utilizar este modo de análisis es necesario especificar el *enlace* que se desea desconectar.

Funciones relacionadas:

```
_analyze.c: analyze_mol()
_analyze.c: analyze_molfgi()
_analyze.c: analyze_molfga()
_analyze.c: analyze_molgp0()
```

Identificación de retronos

Una vez determinado el modo de análisis que se va a utilizar, se procede a identificar los retronos que permitan aplicar una o más de las transformadas que se tienen en la base de datos.

Para ello se utiliza la información contenida en el encabezado de cada transformada. El encabezado especifica el tipo de transformada y las características estructurales que componen el retrón. Éstas pueden ser un grupo funcional, dos grupos funcionales con un camino de longitud definida entre ellos, tipo de anillo, etc. (ver Tabla 5-1). La descripción detallada del formato del encabezado se encuentra en el Apéndice B.

Los algoritmos para las transformadas con retrón pueden ser descritos de manera muy escueta de la siguiente manera:

- **Un grupo funcional.** Para cada grupo funcional presente en la molécula, encontrar y aplicar las transformadas de un grupo funcional que lo requieran.
- **Dos grupos funcionales.** Para cada par de grupos funcionales presentes en la molécula,⁶ encontrar las transformadas que requieran dicho par; en las transformadas que lo requieran, determinar la longitud del camino entre los dos grupos y ver si es la longitud especificada. De ser así, aplicar la transformada.
- **Anillo.** Para cada anillo o sistema de anillos fusionados presente en la molécula, encontrar y aplicar las transformadas de anillos que lo requieran.

Funciones relacionadas:

```
_analyze.c: try_gp1()  
_analyze.c: try_gp2()  
_analyze.c: try_ring()  
_analyze.c: try_mring()  
_analyze.c: try_fga()  
_analyze.c: try_gp0()
```

Aplicación de la transformada

Inicialización → interpretación

La aplicación de una transformada consta de dos etapas. En la inicialización se crea una *copia de trabajo* de la estructura por analizar y se crean todas las variables automáticas usadas por la transformada. Por ejemplo, en una transformada de un grupo funcional se asignan las variables A1, A2, ..., AN a los átomos del primer grupo funcional y se asigna el rating predeterminado. La explicación detallada del uso y significado de las variables predeterminadas se encuentra en el Apéndice B.

⁶ Se puede demostrar fácilmente que hay $N(N-1)/2$ pares de grupos funcionales en una molécula con N grupos funcionales

La interpretación consiste en la lectura secuencial y ejecución de las instrucciones contenidas en el cuerpo de la transformada. La secuencia de ejecución puede ramificarse mediante instrucciones apropiadas, tales como `if ... then`, y `foratom ... next`.

El cuerpo de la transformada tiene dos funciones:

1. Decidir si la transformada es aplicable realizando los chequeos necesarios: presencia o ausencia de ciertos tipos de átomos o grupos funcionales en el entorno de los átomos y enlaces que van a ser afectados. Si la transformada es aplicable, calcular el `rating` final tomando en cuenta las características estructurales del entorno.
2. Aplicar la transformada. Para ello se utilizan instrucciones especiales de ruptura y creación de enlaces, así como de adición y remoción de átomos. Por ejemplo, `breakbond`, `makebond` y `addatom` (ver el Apéndice B).

Funciones relacionadas:

`_execute.c: executegp1()`

`_execute.c: executegp2()`

`_execute.c: executering()`

`_execute.c: executemring()`

`_execute.c: executefga()`

`_execute.c: executefg0()`

`_execute.c: execute()`

Posprocesamiento

Percepción precursores → eliminación de duplicados → evaluación → ordenado

Una vez aplicadas las transformadas, es necesario realizar un posprocesamiento de la lista de precursores generada. La función principal de este posprocesamiento es el cálculo de la simplificación obtenida con cada transformada, para poder comparar y evaluar los resultados.

Primero se realiza la percepción de los precursores. Para ello se utiliza exactamente el mismo algoritmo que detallamos anteriormente para la percepción de la molécula

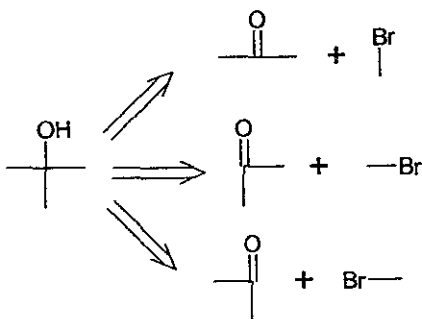


Figura 5-11 Ejemplo de desconexiones que generan resultados redundantes.

objetivo, que consiste en detección y clasificación de anillos, canonización, clasificación de átomos, detección de grupos funcionales y cálculo de complejidad.

Una vez concluida la percepción, es posible comparar las estructuras de los precursores entre sí para detectar duplicados (recordemos que solamente se pueden comparar con facilidad las estructuras canónicas). Los duplicados son eliminados con el fin de no presentar al usuario resultados redundantes. Esto ocurre con especial frecuencia en moléculas con átomos equivalentes. Por ejemplo, si desconectamos cualquiera de los tres enlaces carbono-carbono del alcohol *ter*-butílico, obtenemos exactamente los mismos precursores: acetona y bromometano (Figura 5-11).

Enseguida se procede a evaluar la transformada. La evaluación es el producto de dos factores: el *rating* previamente calculado durante la interpretación de la transformada, y la simplificación, que se calcula entonces. La fórmula utilizada para el cálculo de la simplificación es muy sencilla: se divide la complejidad de la molécula objetivo entre la

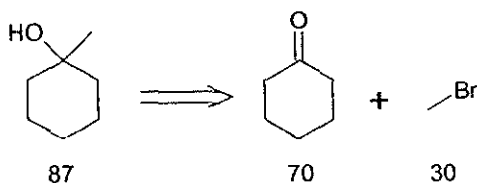


Figura 5-12 Ejemplo del cálculo de simplificación obtenida con una transformada retrosintética. Los números que aparecen debajo de las estructuras son la complejidad de cada una de ellas. En este caso la simplificación fue de $87 / 70 \times 100 = 124$

complejidad del más complejo de sus precursores y se multiplica el resultado por 100. En la Figura 5-12 se muestra un ejemplo de este cálculo. En este caso se divide entre 70, la complejidad de la ciclohexanona, por ser ésta más compleja que el bromometano.

Por último, se ordenan los resultados por valor decreciente de la función de evaluación ($\text{simplificación} \times \text{rating}$). Esto tiene el único fin de mostrar primero al usuario los resultados más prometedores (a juicio del programa).

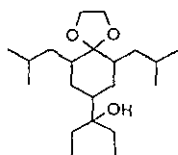
Funciones relacionadas:

`_analyze.c: postprocess_mol_list()`

6. CASOS ESTUDIADOS

En este capítulo presentaré y discutiré un par de ejemplos de moléculas que fueron analizadas retrosintéticamente usando el programa OSET. Esto permitirá al lector darse una idea del tipo de reacciones y grupos funcionales que puede manejar actualmente el programa. Hay que destacar que no siempre es trivial encontrar estas rutas sintéticas: el usuario debe tener una idea acerca de la estrategia que desea utilizar y aplicar las transformadas auxiliares que permitan lograr las desconexiones deseadas.

6.1 Marcianol



Marcianol

Esta molécula imaginaria ha sido usada frecuentemente en la Facultad de Química como ejemplo didáctico.¹ Además de un nombre y una estructura singular, tiene las siguientes ventajas: 1) es una molécula moderadamente compleja ya que tiene anillos, ramificaciones y algunos grupos funcionales;² y 2) al analizarlo retrosintéticamente se

encuentra que se pueden utilizar varias estrategias. A continuación describiré seis rutas diferentes encontradas con ayuda de OSET; ninguna de ellas es perfecta, pero sirven como ejemplo del método de análisis retrosintético.

Ruta 1: Reducción de Birch y alquilación de enolatos

En esta ruta se parte del anisol, una materia prima fácilmente disponible, y se obtiene el marcianol en ocho pasos mediante una síntesis lineal donde se van incorporando otras materias primas con dos a cinco átomos de carbono (Figura 6-1)

¹ Generalmente se le atribuye al profesor Guillermo James el uso de esta molécula, también llamada *moneno*. Otro nombre podría ser 3-(6,10-diisobutil-1,4-dioxaspiro[4.5]dec-8-il)pentan-3-ol.

² El marcianol tiene tres carbonos estereogénicos, pero no los tomaremos en cuenta en nuestro análisis

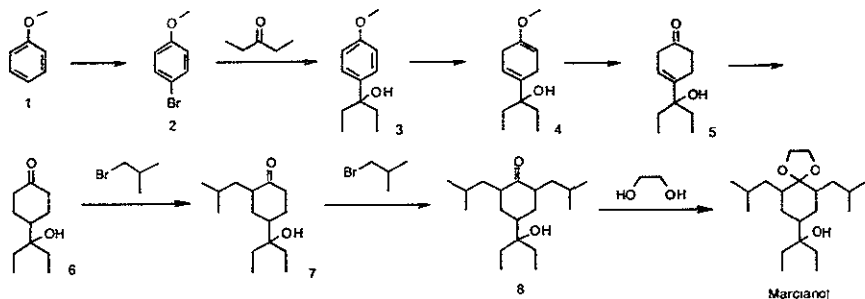


Figura 6-1 Primera ruta propuesta para la síntesis del marcianol. Se parte del anisole, y los pasos principales son una adición de Grignard, una reducción de Birch y dos alquilaciones alfa a carbonilo.

El primer paso es una halogenación por sustitución electrofílica aromática. El anillo del anisole está activado y orienta la sustitución hacia las posiciones *orto* y *para*. Los inconvenientes de este paso serían la posible polihalogenación y la separación de la mezcla de productos.

El segundo paso consiste en la formación de un reactivo de Grignard, seguida por la adición a la 3-pentanona e hidrólisis para dar el compuesto 3. El problema con este paso es que al acidular es altamente probable que se deshidrate el producto, ya que se trata de un alcohol terciario y en posición bencílica. En la práctica podría ser necesario proteger el hidroxilo.

Los siguientes pasos son la reducción de Birch seguida por una hidrólisis para desenmascarar la cetona, y luego una hidrogenación catalítica. Después se procede a alquilar los dos carbonos alfa a la cetona con bromuro de isobutilo. Quizá sería posible realizar estos dos pasos como uno solo. Al ser el isobutilo un grupo voluminoso aumenta la probabilidad de que efectivamente se alquilen ambos carbonos alfa sin que ocurra la alquilación doble en alguno de ellos. En los pasos de alquilación también sería buena idea tener al alcohol protegido, para evitar la formación de un éter no deseado.

Por último se protege la cetona con etilenglicol, obteniéndose el marcianol.

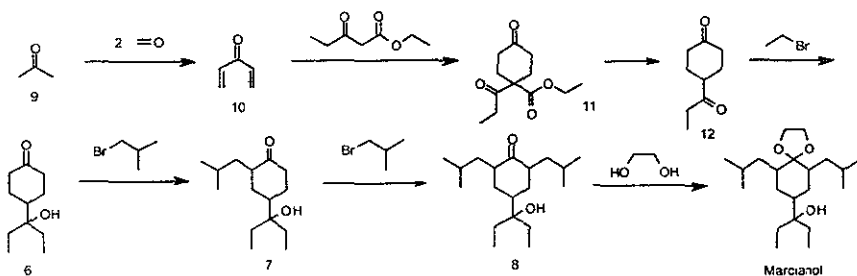


Figura 6-2 Segunda ruta propuesta para la síntesis del marciano. Se parte de acetona y los pasos principales son dos condensaciones aldólicas, una adición de Michael doble, una adición de Grignard y dos alquilaciones de enolatos. Nótese que los tres últimos pasos son iguales que para la ruta 1.

Ruta 2: Adición de Michael

Esta síntesis también es esencialmente lineal (Figura 6-2), y su ventaja con respecto a la primera es que parte de moléculas acíclicas más simples. Todas las reacciones usadas son típicas de los compuestos carbonílicos, por lo que esta síntesis podría ser un buen ejemplo en un curso de esta asignatura.

El primer paso de esta ruta son dos condensaciones aldólicas con deshidratación; el segundo paso es una doble adición de Michael; y el tercer paso es una hidrólisis de éster seguida de una descarboxilación.

El paso que podría ser más problemático es la adición de Grignard al compuesto **12**. En este compuesto se tienen dos cetonas con reactividad semejante, por lo que no se esperaría obtener el producto selectivamente.

Los últimos tres pasos de la síntesis son iguales que para la ruta 1.

Ruta 3: Acilación de Friedel–Crafts y reducción de Birch

En esta síntesis (Figura 6-3), al igual que en la ruta 1, se parte de anisol. Los primeros dos pasos son iguales que en la ruta 1, pero en ésta se utiliza la acilación de Friedel–Crafts en lugar de la alquilación de enolatos. Evidentemente, es necesario realizar la acilación antes que la reducción de Birch.

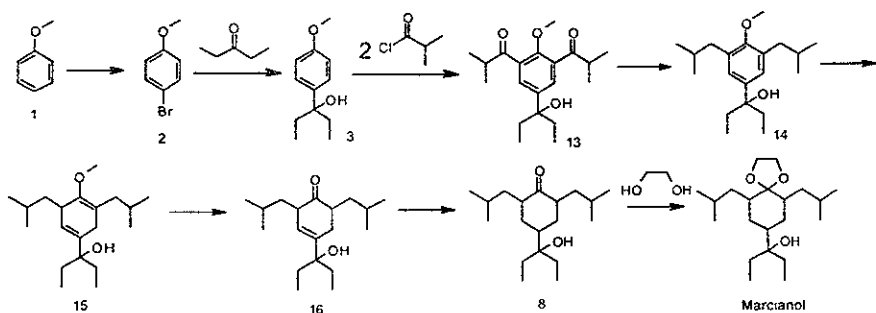


Figura 6-3 Tercera ruta propuesta. Al igual que en la ruta 1, se parte de anisol, pero en este caso se utiliza la acilación de Friedel-Crafts en lugar de la alquilación de enolatos.

El paso clave de esta ruta es la acilación del compuesto 3. Dicho compuesto tiene un grupo activante fuerte que orienta a las posiciones *orto* (que son las posiciones deseadas) y a *para*. Como la posición *para* está bloqueada, toda la sustitución ocurre en *orto*. Hay que tomar en cuenta que después de la primera acilación cambia la reactividad del anillo, ya que el grupo acilo es desactivante; sin embargo, la orientación sigue siendo la misma ya que la posición *meta* con respecto al grupo desactivante es igual a la posición *orto* con respecto al grupo activante.

Después de la acilación se realiza una doble reducción de Clemmensen o de Wolff-Kishner para desoxigenar las cetonas presentes en el compuesto 13. En este caso es preferible la reducción en condiciones básicas (de Wolff-Kishner) para no deshidratar el alcohol. Aun si se protegiera el alcohol, como se propuso para la ruta 1, en las condiciones ácidas de la reducción de Clemmensen podría desprotegerse y deshidratarse.

La reducción de Birch del compuesto 14 no parece tener inconvenientes, ya que el grupo metoxi orienta la reducción de la manera indicada. Este paso es muy similar a la reducción de Birch de la ruta 1, con la diferencia de que se tienen los dos isobutilos sustituyentes unidos al anillo. Después siguen la hidrólisis e hidrogenación, también similares a las de la ruta 1, llegando entonces al compuesto 8. El último paso es igual que en las otras rutas.

Ruta 4: Reducción de Birch y reacción de Wittig

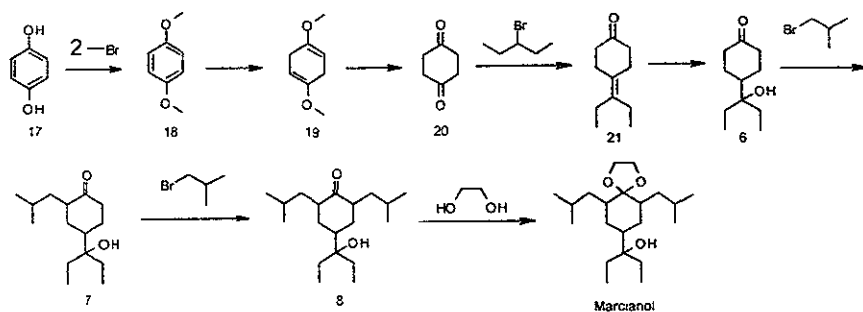


Figura 6-4 Cuarta ruta propuesta para la síntesis del marcianol. Se parte de hidroquinona, y mediante una reducción de Birch y una reacción de Wittig se llega al compuesto 6.

Esta síntesis, mostrada en la Figura 6-4, parte de la hidroquinona, una materia prima ampliamente disponible. Mediante una doble síntesis de éteres de Williamson se forma el 1,4-dimetoxibenceno (18), el cual se reduce con la reacción de Birch y al hidrolizarse da la 1,4-ciclohexanodiona (20). Cabe mencionar que los compuestos 18 y 20 también están disponibles comercialmente, pero decidí extender el análisis retrosintético hasta llegar a la hidroquinona por ser una materia prima más disponible y más barata.

El paso principal de esta propuesta de síntesis es la reacción de Wittig para dar el compuesto 21. Una desventaja importante de este paso es que existe el riesgo de obtener como subproducto la reacción con ambas cetonas.³

La hidratación del compuesto 21 podría dar el compuesto 6; a partir de aquí los pasos restantes son iguales que en la ruta 1. Este es el paso más problemático de esta secuencia, ya que se pretende hidratar regioselectivamente un alqueno tetrasustituido donde no hay gran diferencia entre los dos átomos de carbono involucrados. Probablemente se obtendría una mezcla de productos en proporción cercana a 50:50.

³ Es interesante notar que Aldrich vende el monoetilencetal de la 1,4-ciclohexanodiona, es decir la 1,4-ciclohexanodiona con uno solo de los carbonilos protegido con etilenglicol. En la práctica esa podría ser una buena materia prima ya que resolvería el problema de la doble reacción de Wittig.

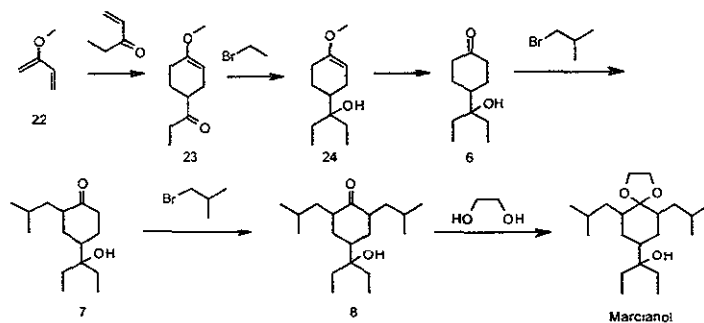


Figura 6-5 Quinta síntesis propuesta para el marcianol. El paso clave de esta ruta es una reacción de Diels–Alder.

Ruta 5: Diels–Alder

Esta ruta tiene la ventaja indiscutible de ser la más corta de las seis que estamos considerando. En seis pasos se obtiene el marcianol a partir de compuestos acíclicos de no más de cinco átomos de carbono, como puede verse en la Figura 6-5.

El primer paso es la reacción de Diels–Alder. En este caso la reacción es bastante favorable, ya que el dieno tiene un grupo electrodonador y el dienófilo tiene un grupo electroattractor. La relación 1,4 (“tipo *para*”) entre dichos grupos en el producto es la que se observa generalmente en este tipo de reacción. De hecho, nuestro caso se parece mucho a uno de los ejemplos típicos que usan los libros de texto para explicar la regioquímica de la reacción de Diels–Alder.⁴

Después de la reacción de Diels–Alder, solamente restan una adición de Grignard y una hidrólisis para llegar al compuesto **6**. Ninguno de estos pasos parece tener mayor dificultad. Después siguen los tres pasos finales ya conocidos desde la ruta 1.

⁴ Véase por ejemplo el libro *Advanced Organic Synthesis* de Carey y Sundberg (1990b), página 286

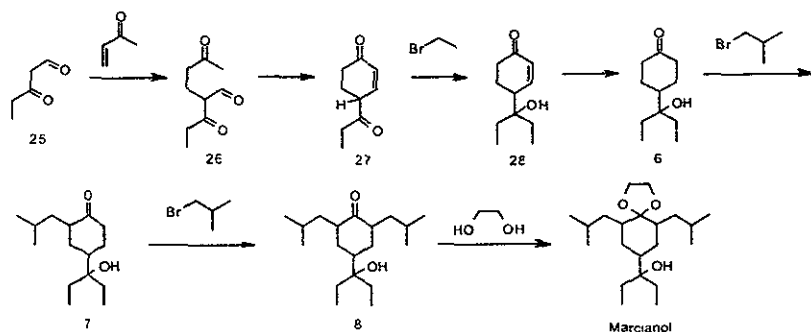


Figura 6-6 Sexta propuesta de síntesis del marcianol. En este caso se parte de una anillación de Robinson.

Ruta 6: Anillación de Robinson

En la Figura 6-6 puede verse la sexta y última propuesta de síntesis del marcianol. Se parte del compuesto dicarbonílico 25. Este compuesto no está disponible comercialmente porque al ser 1,3-dicarbonílico y tener un grupo aldehído sería altamente reactivo y se autocondensaría. Por ello habría que prepararlo en el momento, posiblemente a partir de butanona. Al condensar el compuesto 25 con metil vinil cetona, se espera que se forme el compuesto 27 por anillación de Robinson. Existe un riesgo muy alto de que la autocondensación ocurra predominantemente en lugar de la reacción deseada, pero afortunadamente este paso incierto es el primero de la síntesis y no el último, así que el riesgo que se correría al intentar esta ruta no es tan alto. Además podrían considerarse materias primas alternativas que incluyan un grupo electroattractor que pueda convertirse posteriormente en aldehído. Un ejemplo que viene a la mente es el grupo nitró.

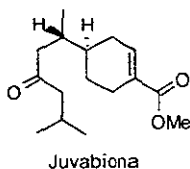
Una vez lograda la anillación, sigue otro paso conflictivo. Se desea lograr la adición de Grignard selectivamente a la cetona acíclica del compuesto 27. En principio parece razonable, ya que la otra cetona presente en la molécula tiene un doble enlace conjugado que disminuye su reactividad. El problema es que el hidrógeno mostrado es bastante ácido y podría reaccionar primero con el reactivo de Grignard. Esto se debe a que está en posición alfa con respecto a una cetona y en posición gamma conjugada con respecto a la otra. Este problema podría resolverse hidrogenando primero el doble enlace; entonces se

LIBRARY
UNIVERSITY OF MICHIGAN
ANN ARBOR MI 48106-1000

obtendría el compuesto **12** (discutido en la ruta 2), lo que haría más difícil lograr una adición selectiva.

Después de la adición de Grignard seguida por hidrogenación o viceversa, se llega finalmente al compuesto **6**. Los siguientes tres pasos son los mismos de la ruta 1.

6.2 Juvabiona



El marcianol es una molécula imaginaria y por lo tanto no hay síntesis reportadas para comparar. Esto es a la vez una ventaja y una desventaja, ya que por un lado se evitan los prejuicios de las síntesis reportadas, pero por otro lado no hay síntesis probadas para comparar. Por eso decidí incluir la juvabiona como segundo ejemplo.

La juvabiona es una sustancia producida por las coníferas, similar a una hormona de un insecto perjudicial; por ello, podría utilizarse para el control natural de las plagas, evitando que los insectos alcancen la madurez (Warren, 1978).

La juvabiona tiene dos centros quirales, por lo que tiene cuatro estereoisómeros (dos pares de enantiómeros). Aunque ya ha sido informada la síntesis estereoselectiva de la juvabiona (la primera es de Pawson, 1970; la más reciente es de Bergner, 2000), discutiré solamente dos rutas de síntesis no estereoselectivas publicadas ya que OSET no toma en cuenta la estereoquímica. En este ejemplo procederé al revés que en el anterior: primero presentaré la síntesis publicada y luego discutiré si es reproducible por OSET.

Síntesis de Ayyar y Rao (1967)

Esta síntesis de once pasos (Figura 6-7) parte de la *p*-metoxiestirilisobutil cetona **29**. Mediante una adición conjugada de CH₃MgBr se obtiene el compuesto **30**, el cual es reducido al alcohol **31**. La reducción de Birch da el compuesto **32**, el cual es hidrolizado para desenmascarar la cetona en el compuesto **33**. El alcohol libre se protege por acetilación para dar el compuesto **34**; posteriormente se procede a hidrogenar el enlace doble C=C para dar el compuesto **35**. El compuesto **35** se convierte en la cianhidrina **36**, la cual se deshidrata para dar **37**, se hidroliza para dar el ácido carboxílico **38**, y se oxida para dar **39**. Al esterificar el compuesto **39**, llamado *ácido todomáutico* (debido a que fue

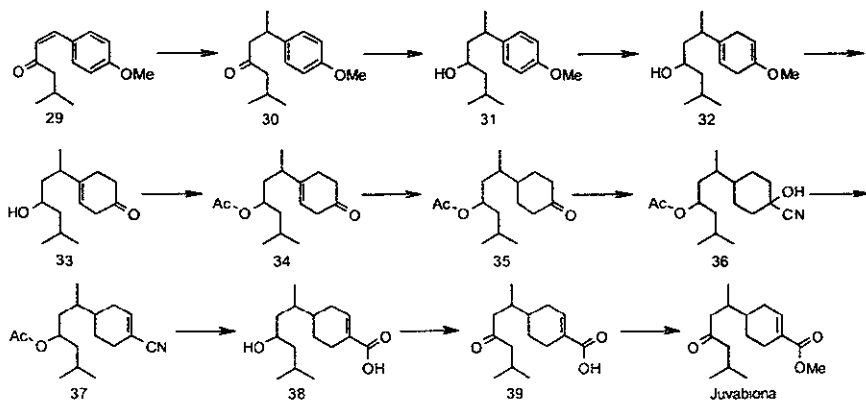


Figura 6-7 Síntesis de juvabiona de Ayyar y Rao.

encontrado en un árbol japonés llamado *todomatsu*), se obtiene la juvabiona, junto con sus tres estereoisómeros. Los intentos de los autores para separar los isómeros fracasaron (Ayyar, 1967).

El análisis de la juvabiona con el programa OSET reveló que sí es posible encontrar esta ruta sintética. Todas las reacciones utilizadas se encuentran en la base de datos de OSET, ya que son bastante conocidas y suelen ser estudiadas en la licenciatura. Sin embargo, sería raro encontrar exactamente esa ruta en un análisis retrosintético inicial, ya que los pasos de protección y desprotección del alcohol suelen dejarse para una etapa posterior del diseño de la síntesis. Otro detalle que hay que mencionar es que con OSET se encuentra la doble hidrólisis de 37 para dar 38 como si fueran dos pasos separados, mientras que en la práctica son simultáneos. Eso ocurre en general con todas las reacciones “múltiples”, tipo dominó (*tandem*), etc.

Hay que hacer énfasis en que actualmente OSET no identifica las interferencias entre grupos funcionales; esa labor se le deja al usuario. Por ello, el programa no tendría inconveniente alguno en proponer una ruta donde el grupo carbonilo presente en la materia prima 29 se dejara intacto a lo largo de toda la síntesis. Con ello se obtendría una síntesis mucho más corta pero que en la práctica no funcionaría. Por ejemplo, habría problemas en el paso de formación de la cianhidrina ya que habría dos cetonas en la molécula.

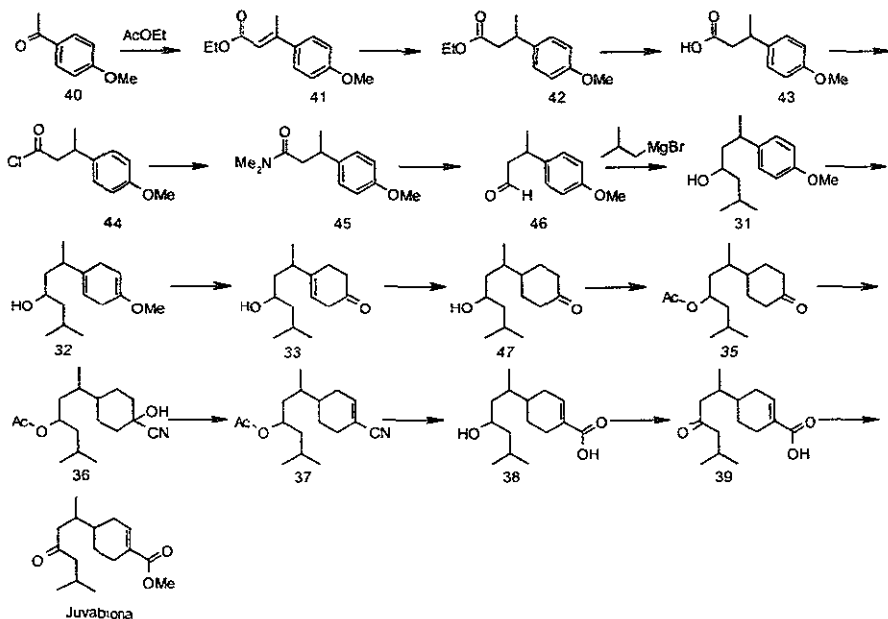


Figura 6-8 Síntesis de juvabrona de Mori y Matsui.

Síntesis de Mori y Matsui (1968)

Esta síntesis tiene 16 pasos y parte del *p*-acetil anisol **40**, un compuesto más simple que el utilizado por Ayyar (Figura 6-8). La parte clave de la síntesis es esencialmente igual a la anterior ya que también se basa en una reducción de Birch; los últimos cinco pasos son idénticos.

El compuesto **41** se obtiene a partir de **40** mediante una reacción de Reformatsky. Posteriormente se hidrogena el doble enlace para dar **42**. Sigue entonces una serie de transformaciones cuyo fin es reducir el éster a aldehído. Se pasa por el ácido **43**, el cloruro de acilo **44** y la amida **45** para dar finalmente el aldehído **46**. Quizá con reactivos más selectivos como el DIBALH se podría resumir esta secuencia a un solo paso (ver por ejemplo March, 1992, p. 447). Los autores usaron hidruro de trietoxialuminio como agente reductor.

Una adición de Grignard al aldehído da como producto el alcohol **31**, que ya habíamos visto en la síntesis de Ayyar. A partir de ahí la síntesis es igual excepto en que al pasar de **33** a **35** se prefirió invertir el orden: primero se hidrógeno el alqueno y luego se acetiló el alcohol. Los autores de esta síntesis sí encontraron un método para separar los dos pares de enantiómeros obtenidos finalmente. Para ello sintetizaron la semicarbazona y la separaron por cristalización.

Esta ruta se pudo reproducir casi completamente en el programa OSET. Una diferencia, además de las mencionadas para la síntesis de Ayyar, fue que OSET no manejaba la reducción de amidas a aldehídos. Sin embargo, sí maneja la reducción de cloruros de acilo a aldehídos, por lo que se obtuvo una ruta más corta por un paso. Otra diferencia fue que en el primer paso OSET no propuso la reacción de Reformatsky, pero sí propuso la reacción de Wittig, que es una transformación equivalente;⁵ también propuso una condensación aldólica que sería una transformación muy parecida.

De esta síntesis también se concluye que aunque OSET difícilmente propone rutas de síntesis “listas para usarse”, puede ayudar a proponer un esquema que luego deberá ser afinado con el juicio de un buen químico.

⁵ Se entiende por transformación el cambio estructural neto sin tomar en cuenta los reactivos e intermedios.

7. APLICACIONES EDUCATIVAS

7.1 Antecedentes históricos

Desde que se empezaron a usar las computadoras en síntesis han aparecido varios artículos o comunicados en el *Journal of Chemical Education* describiendo el uso de las computadoras en enseñanza de la síntesis (Smith, 1971; Orf, 1975; Feldman, 1976; Stolow, 1980; Sandel, 1981; Barone, 1985; Bertrand, 1986).

Los programas utilizados pueden clasificarse en dos tipos: 1) programas sofisticados de síntesis por computadora cuyo uso principal es en investigación pero han sido aplicados en la enseñanza (los artículos de Orf, Stolow y Bertrand pertenecen a esta categoría); 2) programas relativamente sencillos desde el punto de vista de la síntesis pero diseñados especialmente como herramientas de enseñanza (Smith y Barone). Hay un caso digno de mención que no pertenece a ninguna de estas categorías. El programa ORGSYN descrito por Sandel fue el producto de un seminario de síntesis por computadora, en lugar de ser una herramienta del curso. Los estudiantes que participaron aprendieron mucho sobre síntesis *mientras desarrollaban el programa*. Sin embargo, no se llegó a la etapa de implementación.

El caso más antiguo es el de Smith (1971), en la Universidad de Illinois: desarrolló un programa educativo con interfase de texto, con una base de datos de 53 reacciones específicas y 134 reactivos. El programa no realizaba un análisis propiamente dicho, sino que funcionaba básicamente como un examen de opción múltiple. Este programa no corresponde exactamente con la idea de síntesis orgánica asistida por computadora que se presenta en esta tesis, pero lo incluyo por razones históricas.

El profesor Orf (1975) tomó el LHASA y lo utilizó con un grupo piloto de 15 estudiantes voluntarios. LHASA fue diseñado para ser utilizado por profesionales, no por estudiantes; sin embargo, su interfase gráfica (toda una novedad en aquel momento)

facilitaba mucho la comunicación al nivel de permitir que se utilizara para la enseñanza. El autor concluyó que “el potencial de LHASA como una herramienta educativa para la síntesis orgánica había quedado demostrado”. El experimento otorgó confianza a los alumnos para trabajar con problemas de síntesis no triviales, aun sin ayuda de la computadora.

En el artículo de Stolow (1980), se usó una versión del LHASA modificada especialmente para la enseñanza. Las principales modificaciones fueron: 1) se eliminó gran parte de las reacciones, dejando solamente las de interés para ese curso; y 2) se agregaron secciones interactivas donde se puede evaluar la capacidad del alumno de identificar los grupos funcionales, enlaces estratégicos, transformadas apropiadas, etc. La respuesta de los estudiantes fue muy positiva.

Los programas de Barone (1985) eran relativamente simples y funcionaba en una Apple II+. Había tres programas basados en una matriz de interconversión que incluía 36 compuestos genéricos (por ejemplo, RCHO) y 100 reacciones de un paso que transformaban unos compuestos en otros, usando 38 condiciones de reacción diferentes. El primero de los programas, llamado REVIEW, pedía al estudiante todos los precursores y condiciones de reacción para un compuesto escogido al azar. El programa ONESTEP escogía dos compuestos al azar y preguntaba al estudiante si la interconversión es posible y bajo qué condiciones. Por último, el programa MULTISTEP escogía dos compuestos al azar y generaba automáticamente todas las síntesis de un paso o más que los relacionaran. Entonces pedía al estudiante una ruta de síntesis y la comparaba con las que había generado. Estos programas tampoco son propiamente de síntesis asistida por computadora, pero son dignos de mención por ser los primeros escritos para microcomputadora.

El artículo de Bertrand (1986), relata la aplicación del programa SOS en la enseñanza de la síntesis orgánica. SOS era un programa de investigación que fue adaptado para funcionar en una microcomputadora Apple II+. Algo interesante de este artículo es que, como parte del ejercicio, después del análisis retrosintético de la molécula objetivo (\pm)ar-turmerona (Figura 7-1), el grupo discutió las rutas posibles con ayuda del instructor y finalmente llevaron una de ellas a la práctica. La síntesis que realizaron ya había sido publicada antes, pero los estudiantes no la conocían.

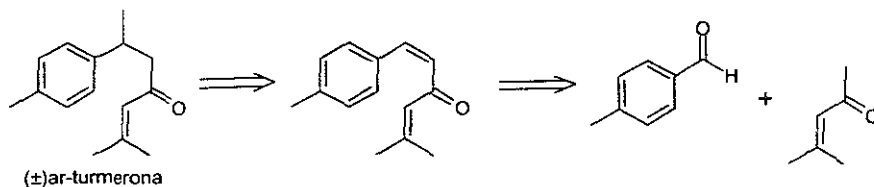


Figura 7-1 Análisis de la (±) ar-turmerona llevado a cabo por el grupo de Bertrand

Los artículos más recientes son los de 1986, y solamente los dos últimos (Barone y Bertrand) describen programas para microcomputadoras (en ambos casos, la Apple II+). Todos los anteriores requieren computadoras más sofisticadas y caras. Esto se debe sin duda a que las microcomputadoras surgieron apenas en 1975, y eran muy lentas en un principio. Sin embargo, las PCs de ahora son del orden de 1000 veces más rápidas que las de 1986, por lo que ahora es posible escribir programas sofisticados para la PC.

Puede verse que en todos los casos las experiencias relatadas por los autores arriba mencionados fueron positivas. Sin embargo, como la mayoría de los programas requerían recursos computacionales poco disponibles, el impacto de este recurso pedagógico fue muy limitado. No pude encontrar ningún artículo posterior a 1986 sobre el tema, por lo que parecería que ha caído en el olvido. Sin embargo, en años más recientes han aparecido algunos programas para Windows y Macintosh que posiblemente se utilizan actualmente en algunas universidades, sin que haya habido publicaciones al respecto.

En esta situación se encuentra SynTree, programa que tiene un costo de \$200 USD y que funciona en Windows.¹ Otros programas de investigación como CAMEO y SynGen han sido adaptados para funcionar en la Macintosh, y HOLOWIN es un ejemplo de un programa para investigación que está disponible para Windows.

El objetivo de esta tesis, como se dijo en un principio, es rellenar este hueco con un programa útil para enseñanza, gratuito y distribuido de manera libre. Esto incluye la distribución del código fuente para facilitar las correcciones y modificaciones al programa.

¹ En la opinión del autor de esta tesis, el programa SynTree deja mucho que desear, especialmente considerando su costo. Además de tener una cantidad generosa de *bugs* (errores de programación), es difícil encontrar rutas sintéticas útiles usando este programa. Para mayor información sobre este programa, ver su página en <http://www.trinitysoftware.com/trinity2/orgchem/syntree.htm>

7.2 Posibles aplicaciones del programa

OSET en enseñanza

El programa OSET podría ser una herramienta útil en la enseñanza de la síntesis orgánica para estudiantes tanto de nivel básico como avanzado. En el caso particular de la Facultad de Química de la UNAM, podríamos clasificar las asignaturas de química orgánica de la licenciatura en química como se muestra en la Tabla 7-1.

Asignatura	Nivel
Introducción a la Química Orgánica (1335)	básico
Compuestos con C, H, y X (1435)	básico
Compuestos con C, H, O, N y S (1535)	intermedio
Compuestos carbonílicos (1635)	avanzado
Compuestos heterocíclicos (1735)	avanzado

Tabla 7-1 Asignaturas de química orgánica de la FQ-UNAM. Todas ellas tienen duración de un semestre con aproximadamente 48 horas de teoría y 56 de laboratorio.

La ventaja de las asignaturas avanzadas, especialmente *Compuestos carbonílicos* (1635), es que se estudian muchas reacciones de formación de enlaces carbono-carbono como las que utiliza el programa OSET (o cualquier análisis retrosintético clásico). Por ello, parecería que este es el semestre ideal para estudiar el método de análisis retrosintético, y es donde podría utilizarse la computadora. De esta manera se reduciría el problema de que los estudiantes “vean los árboles (las reacciones) pero no el bosque (la síntesis)”

En las asignaturas 1435 y 1535 se estudian principalmente reacciones de intercambio de grupos funcionales, por lo que parecería difícil estudiar síntesis interesantes en ese momento. Sin embargo y precisamente por ello, ese sería un buen momento para intentar el estudio de los principios generales del análisis retrosintético aun sin conocer bien todas las reacciones sintéticas. Esto podría ayudar a los estudiantes a poner en perspectiva el uso de las reacciones de intercambio de grupos funcionales y comprender su utilidad como parte de un esquema de síntesis más amplio. El estudiante no necesitaría conocer la reacción sintética ya que de eso se encargaría la computadora, sin duda sería difícil pedirle al alumno que discuta detalladamente la viabilidad de una transformada dada si desconoce

el mecanismo y las condiciones de reacción, pero sí podría evaluar su posible utilidad para la simplificación de la molécula objetivo.

En la *Introducción a la Química Orgánica*, podría intentarse enseñar al alumno las nociones de la síntesis orgánica, sin detalle de las reacciones porque casi no se estudian reacciones en este curso. Sin embargo, habría que tener cuidado y en todo caso realizar esta actividad al final del semestre, porque al principio los estudiantes todavía no dominan la notación estructural, los grupos funcionales y otros conceptos básicos fundamentales.

En resumen, podría usarse el programa OSET en diversos cursos² siempre y cuando se tome en consideración el nivel de los estudiantes y se diseñen los ejercicios de manera adecuada. Esto se logra de tres maneras:

1. Adecuando el nivel de análisis que se le pide al alumno (detalles de reacciones vs exploración general).
2. Escogiendo apropiadamente los ejercicios, es decir, las moléculas por analizar.
3. Editando la base de datos de transformadas para que sólo se utilicen las reacciones que el profesor considere adecuadas.

Para los alumnos de *Compuestos Carbonílicos* podrían realizarse ejercicios similares al del marcianol (ver el capítulo anterior): no sólo se le pediría al estudiante que proponga las rutas de síntesis, sino también que las discuta con base en sus conocimientos de mecanismo y reactividad.

En semestres un poco más básicos se esperaría que el alumno *explore* el árbol sintético (de ahí el nombre del programa OSET, *Organic Synthesis Exploration Tool*). No se le pedirían detalles sobre reacciones que no conoce, pero sí se esperaría que se diera una idea de la cantidad de posibilidades de síntesis diferentes, es decir, del tamaño del árbol sintético. Además habría que escoger los problemas cuidadosamente y editar la base de datos de transformadas para que en la medida de lo posible el análisis se concentre en las reacciones conocidas

Una ventaja muy importante del uso de síntesis no triviales en la enseñanza de la química orgánica es que se facilita la integración entre los cursos: se obliga al estudiante y al profesor a recordar que las reacciones que se estudian en un curso no son independientes

² Claro está que no sería recomendable usarlo en todos los niveles porque podría resultar monótono para los estudiantes

ni se usan de manera aislada. Entre los estudiantes suele existir la política de que una vez terminado un semestre se olvidan de todo y vuelven a empezar desde cero. Esta política es reforzada inconscientemente por algunos profesores que imparten *su* materia de manera aislada, sin enfatizar la relación que tiene con otras, y son renuentes de preguntar en tareas y exámenes temas vistos en el curso anterior.

7.3 Pruebas de campo

Aunque sin duda sería deseable realizar un experimento educativo para determinar la utilidad del programa OSET en la enseñanza de la química orgánica, considero que ello queda fuera del objetivo de esta tesis ya que podría hacerse una tesis completa sobre el tema. Solamente se realizó una pequeña prueba informal con el grupo de *Desarrollo y Optimización de Procesos de Síntesis II*. Esta es una asignatura optativa programada para el último semestre de la licenciatura en Química de la Facultad de Química de la UNAM. La prueba fue realizada en septiembre del año 2000.

El grupo era de cuatro estudiantes—muestra de dudosa representatividad—y se utilizó como ejercicio de práctica el análisis del marciano1. Las observaciones y conclusiones fueron las siguientes:

- El programa es razonablemente fácil de aprender a usar, ya que en menos de una hora lo podían usar de manera competente.
- El applet de dibujo de estructuras no es tan fácil de usar como se deseaba, ya que utiliza algunas convenciones distintas de las de los programas de dibujo de estructuras más conocidos. Sin embargo, también puede dominarse en un tiempo relativamente corto. Aunque existen en el mercado otros dos applets de dibujo de estructura gratuitos, este era el único que tenía las características necesarias para poder comunicarse correctamente con el programa OSET.
- El programa OSET tenía algunos *bugs* (errores de programación), pero ya han sido corregidos. Sin duda habrá más errores aún no detectados pero, gracias a que distribuimos el código fuente, cualquier otro programador que los encuentre podrá corregirlos.

- No en todas las computadoras que fueron probadas en esta y en otras ocasiones ha podido correr satisfactoriamente el applet OSET. Esto se debe generalmente a *browsers* mal configurados u obsoletos. Este es un problema desafortunado, considerando que entre las metas de diseño del programa está la portabilidad, pero es razonable esperar que con el paso del tiempo predominen las versiones recientes de *browsers* compatibles.
- La computadora no le “resuelve la tarea al alumno”, al menos si la tarea no es trivial. No se observó que los estudiantes encontraran una solución óptima de manera inmediata y sin esfuerzo, sino que tenían que explorar efectivamente diversas posibilidades. Esto es precisamente lo que se desea, ya que algunos colegas entrevistados anteriormente habían expresado su preocupación de que “la computadora le resuelva la tarea al alumno y el alumno ya no piense”. Esta preocupación, aunque es legítima en cierta manera, es similar a la preocupación que existió cuando se inventó la imprenta o la calculadora; Finalmente, no es posible negar las innovaciones técnicas, sino que la docencia debe adaptarse a ellas.

8. CONCLUSIONES

Aunque ningún programa de computadora se termina realmente y OSET no es la excepción, creo que OSET ya está listo para ser utilizado. En ese sentido puedo decir que se cumplió el objetivo principal de este trabajo. Como se vio en los Capítulos 6 y 7, el programa OSET ha demostrado ser razonablemente fácil de usar y parece ser útil para su propósito: permitir una exploración más amplia del árbol retrosintético, especialmente a los estudiantes que no se saben todas las reacciones necesarias de memoria.

OSET cuenta con cerca de un ciento de transformadas retrosintéticas en su base de datos, en el momento de redactar esta tesis.¹ Esto equivale a la mayor parte de las reacciones estudiadas en la licenciatura en Química. Además, gracias al diseño abierto del programa, es fácil agregar nuevas transformadas si éstas se pueden clasificar en uno de los tipos cubiertos por el lenguaje. Hay que reconocer, sin embargo, que existen reacciones que no se pueden agregar fácilmente con el detalle que pudiera desearse debido a las limitaciones en el lenguaje usado para describirlas. Otra de las limitaciones que enfrenta actualmente OSET es que no toma en cuenta para nada la estereoquímica; ésta podría ser una línea de desarrollo futuro.

Queda pendiente otra posibilidad de desarrollo importante, pero que es independiente del propósito educativo del programa: un algoritmo de búsqueda automática que permita la generación de rutas de síntesis completas y que las compare para proponer al usuario las mejores opciones. Es decir, convertir a OSET en un programa *no interactivo*, como se conocen en la jerga de la síntesis orgánica por asistida por computadora. Esta conversión requeriría mucho trabajo tanto a nivel conceptual (¿cómo hacer que la computadora evalúe y compare rutas de síntesis, dando resultados similares a los que propondría un químico?), como de programación (¿cómo optimizar el proceso, ya que el árbol de posibilidades es virtualmente infinito para moléculas grandes?). OSET fue

¹ La lista de transformadas se encuentra en el Apéndice C

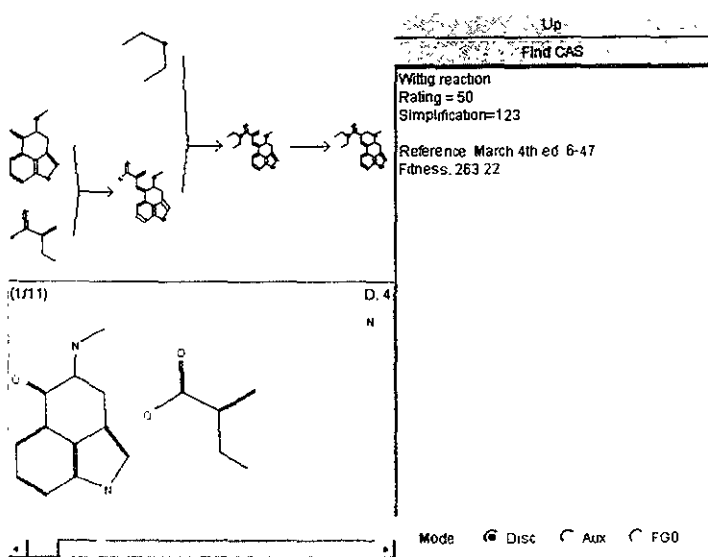
diseñado pensando que algún día podría dársele este uso; para ello puede aprovecharse la capacidad de funcionar en red y algunas funciones de evaluación que están aún en etapa de pruebas.

Finalmente, regresando al uso educativo propuesto para el programa, hay que enfatizar que para que tenga verdadero valor, es necesario que la gente lo use. Todavía queda por delante la promoción de su uso en las aulas de la Facultad de Química. Para promover su uso en el resto del mundo, se creó una página de internet en <http://litio.pquim.unam.mx/osef/>, la cual ha recibido más de 1,900 visitas de 69 países distintos desde el 10 de abril de 2000 hasta la fecha (marzo de 2001).

A. MANUAL DEL USUARIO

Nota: Este apéndice está escrito en inglés porque es el documento de ayuda que aparece en la página de internet junto con el programa OSET. La dirección de este documento en la red es <http://litio.pquim.unam.mx/oset/help.html>

Here is a screen shot of the OSET applet, followed by a description of its components. The target molecule in this example is LSD.



A.1 Upper panel (tree panel)

In this panel you can visualize the entire synthetic tree for the synthesis that is currently under consideration. When the tree is large the molecules may become illegible (especially if the screen resolution is small; we recommend 1024 x 768 or better), but the idea is to get an overview of the structure of the tree, not of each molecule. When possible, the

heteroatoms are indicated by their chemical symbols, else they are shown as color dots according to the following code:

O: red
N: blue
Br: brown
Cl: green
I: purple

A.2 Lower panel (precursors panel)

In this panel you can visualize the precursors from the last retrosynthetic step, as well as unanalyzed precursors from previous steps. On the left side of this panel you can see the precursors from the last step. They are relatively big and with black lines (if a precursor appears with green lines it is because it has been analyzed previously). On the right side of the panel are the *orphans*, precursors found on previous steps that were left unanalyzed. They appear here to remind the user that they must be analyzed later unless they are considered as available starting materials. The orphans appear smaller than the other precursors and with gray lines. If an orphan has been analyzed previously, it appears red.

When a compound is found in our “starting materials catalog”, the name appears under its structure, like diethyl amine in the example. The catalog contains currently 80 compounds; not much, but they are the most common acceptable starting materials in textbook problems and exercises (mostly four carbon atoms or less).

The **scrollbar** under this panel is used to view alternative precursors for the current target. On the upper left corner you can see how many results were found for the current target. In the example, we are viewing the first result out of eleven. The results are sorted in decreasing order of preference according to our evaluation function, but the user does the best evaluation, so please check all the results. The number on the upper right indicates the depth of the analysis, that is, the number of analysis steps.

To analyze a molecule, just click over the structure. The mouse cursor should change to a hand when it is over a clickable structure, and during the analysis it should change to a hourglass or equivalent. The analysis usually takes a few seconds, but the delay depends more on the speed of the network connection than anything else. When analyzing a structure, remember to select the desired analysis mode (see below).

A.3 Right panel (information panel)

In this panel you can read the information regarding the last analysis step. The following information is available:

- **Transform name:** In this case the transform corresponds with the Wittig reaction.
- **Rating:** An evaluation of the transform on chemical grounds. The rating should consider factors such as yield, cost, selectivity, reliability, etc., but currently the rating function is very crude.
- **Simplification:** A measure of the simplification (decrease in molecular complexity) achieved by the use of this transform. Notice that the results are sorted by the product Rating · Simplification.
- **Comments:** This includes more information about the reaction, including reaction conditions and other comments, and the reference. In this example the only information provided is a reference.
- **Fitness:** This function is under development and it is not very useful at this time, but it should be a global evaluation function which takes into account not only the rating and simplification, but also the structure of the tree.

Up button

Click on this button to go up one level in the retrosynthetic tree. When you are on the first level (i.e., the main target molecule) this button is grayed.

Find CAS button

Use this button to get the names, SMILES, and CAS registry numbers of all the compounds in the precursors panel. We currently use a catalog with more than 80,000 compounds. If a molecule is not found, it returns “unknown”.

Mode selector

This radio button is used to select between three modes of analysis: disconnective, auxiliary, and FGO

- **Disconnective:** This is the default mode, which uses only transforms which break skeletal bonds (mainly C-C bonds). This mode applies to the entire structure of a molecule. This class of transforms include GP1 and GP2 transforms, which are keyed by one or two functional groups in the molecule, and ring transforms.
- **Auxiliary:** This mode uses transforms which do not simplify the target structure, such as *functional group interchanges* (FGI) and *functional group additions* (FGA). These transforms are used to obtain the full retron for a *simplifying* (i.e., disconnective) transform. **To use an auxiliary transform, you must click on an atom**, because otherwise the number of results would be unwieldy. Only transforms which affect the functional group which contains the selected atom are used.
- **FG0 (that's a zero!):** These are disconnective transforms that are not keyed by functional groups. This includes some coupling reactions, such as the Wurtz coupling. **To use an FG0 you must click on a bond.**

Important note: If you analyze a molecule, then go up, and then analyze it again using a different mode (or a different atom or bond), the new results are appended after the previous results. So if it seems like the results are not new, try moving to the end of the list.

B. MANUAL DE PROGRAMACIÓN

Nota: Al igual que el anterior, este apéndice está escrito en inglés. Esto se debe a que, al estar destinado OSET a una audiencia internacional, el programa está en inglés al igual que su documentación. Este documento puede encontrarse en internet en la dirección <http://litio.pquim.unam.mx/oset/doc.html>.

B.1 Functional group description language

Functional groups are described in the file `funcgrp.chm`. Each entry consists of the following:

1. Group name. This identifier must be unique and cannot be a reserved word. This means that it is not possible to use an atom type as a name. The name must begin with a letter or underscore, followed by any length of letters, numbers or underscores.
2. Pattern string. This is described below.
3. Complexity.

For example, the line describing carboxylic acids might look like this:

```
ACID = "C(=O O([H]) [CH])", 10
```

The pattern string consists of several “hard” and “soft” atoms. The hard atoms *must* be present in the structure and are automatically numbered from left to right and assigned the default variables $A_1 \dots A_N$. Soft atoms may be several atoms that must match a list of elements. **A** is a hard atom and **O** is a soft atom.

description = “A”

$A = bS(A . O)$

$O = b\{S\dots\}^{n-m}$ or $b\{S\dots\}^n$

S = chemical element symbol or /chemical element symbol

b = = | # | ~

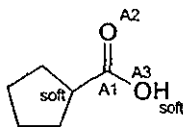
Fields in italics are optional. Notice that the above definition is recursive.

If an element symbol is lowercase it represents an aromatic atom, and if it is preceded by a slash it represents an aliphatic atom. If it has normal case it represents an aromatic or aliphatic atom.

The pattern must begin with a hard atom, which is called the *root atom*, optionally followed by a neighbor list in parentheses. The importance of the root atom is that it is used to calculate distances between pairs of functional groups.

The neighbor list is a list of zero or more hard atoms, optionally followed by a soft atom. The soft atom is enclosed by brackets and optionally followed by a number or number range. Double and triple bonds are represented by = or #, respectively.

In the carboxylic acid example we have as the root atom the carboxyl carbon (numbered A1), followed by: an oxygen bonded to the carbon with a double bond (A2), an oxygen with a single bond (A3), which in turn is followed by a soft hydrogen, and finally a carbon or hydrogen joined to the root atom. Why did we use a soft hydrogen instead of a hard hydrogen? Actually, the only difference is that as a soft hydrogen that atom will not be numbered. Since all hydrogen atoms in our program are implicit, it does not make sense to number them.



Let us look at another example: the alcohol.

ALCOHOL = "C(O([H]) [CH]^3)", 10

Here the carbon is the root atom, and its neighbors are the oxygen (which in turn is bonded to hydrogen) atom and three soft atoms, each of which may be either carbon or hydrogen. Notice that this definition rules out enols, gem-diols, α -halohydrins, and phenols, among others. We decided that this restrictive definition was better from a synthetic point of view. For example, even if an enol may be regarded as a kind of alcohol, its reactions and syntheses are different enough to regard it as an independent functional group.

Here is the complete list of special characters used in functional group descriptions:

- [] Encloses soft atoms. Between the brackets there must be a list of one or more chemical elements. The number of soft atoms may be optionally specified using ^n after the closing bracket.
- ^ Must be followed by a number or a range of numbers. Ranges consist of two numbers separated by a hyphen and enclosed in curly brackets. Numbers may only be used for soft atoms.
- { } Encloses a range.
- () Encloses the neighbor list of an atom. Only hard atoms may have neighbor lists. There may be parentheses inside parentheses, generating a tree-like structure. The neighbor list can have zero or more hard atoms, optionally followed by one soft atom.
- Separates the lower limit from the upper limit of a range.
- = Double bond. Must be immediately before a hard atom.
- ~ "Don't care" bond. Represents any bond.
- # Triple bond. Must be immediately before a hard atom.
- " " Encloses the pattern string.
- / Denotes that the following atom must be aliphatic.

B.2 Transform Description Language

General notes

- The language is case-insensitive.
- The transform database must begin in the file `rxns.chm`, which must be in the working directory.
- More files may be included using the include construct:

```
include "filename"
```

The effect of inclusion is the same as copying the included file and pasting it at the place of the include statement. A file may be included more than once, and it is also possible to include files within included files, up to a maximum depth of two levels.

- The interpretation of the transform database depends also on two other files which are parsed previously: `funcgrp.chm` and `ring.chm`. For details about the structure of

```
.rxn
name "transform name"
type GP1 { GP2 | RING | FGI1 | FGI2 | GP0 | FGA
[G1 functional group list]
[G2 functional group list]
[ringtype ringtype]
[path pathlength]
[gprec precursor group]
[rating =rating]
[link "url"]
[.conditions]
[.comments
"comment"
"comment"
...}

.start
[statement
statement
statement
.]
```

Figure 1 General transform structure

```

.rxn
name "transform name"
type GP1
G1 functional group list
[rating =rating]
[link "url"]
[.conditions]
[.comments
"comment"
"comment"
...]

.start
[statement
statement
statement
...]
```

Figure 2 GP1 transform structure

```

.rxn
name "transform name"
type GP2
G1 functional group list
G2 functional group list
path pathlength
[rating =rating]
[link "url"]
[.conditions]
[.comments
"comment"
"comment"
...]

.start
[statement
statement
statement
...]
```

Figure 3 GP2 transform structure.

these files, read the appropriate sections.

- There are two types of comments: line comments, which are denoted by a semicolon (;) and extend to the end of the line; and block comments, which are enclosed between a slash-asterisk (/*) and an asterisk-slash (*). Block comments may not be nested, but line comments may be within block comments.

Each transform comprises two parts: a header and a body. The header begins with `.rxn`; the body begins with `.start` and ends at the beginning of the next transform (that is, at the next `.rxn`). Figure 1 shows the general structure of a transform. In this figure, optional fields are enclosed by square brackets [], and multiple-choice fields are denoted by vertical bars.

Header

The header has the following items:

```
.rxn
```

Marks the beginning of a new transform. Notice the dot before `rxn`.

```
name "transform name"
```

```

.rxn
name "transform name"
type FGI1
G1 functional group list
gprec functional group list
[rating =rating]
[link "url"]
[.conditions]
[.comments
"comment"
"comment"
...]

.start
[statement
statement
statement
...]

```

Figure 4 FGI1 transform structure

```

.rxn
name "transform name"
type FGI2
G1 functional group list
G2 functional group list
path path length
gprec functional group list
[rating =rating]
[link "url"]
[.conditions]
[.comments
"comment"
"comment"
...]

.start
[statement
statement
statement
...]

```

Figure 5 FGI2 transform structure

The *transform name* must be enclosed by double quotes. The name is a string that is displayed to the user to identify the transform. It is case sensitive and it need not be unique.

type transform type

The *transform type* must be GP1, GP2, RING, FGI1, FGI2, FGA, or GP0. The first three types are disconnective transforms, and are used always when possible. The other types are not called automatically; they are called only when the user selects the appropriate transform type and clicks on a particular atom or bond. See Figures 2-8 for general examples about the structure of each transform type.

- GP1 - one-group transform. The retron is a single functional group.
- GP2 - two-group transform. The retron consists of two functional groups separated by a path of fixed length.
- RING - the retron is a ring, as defined in the file `ring.chm`.
- FGI1 - one-group functional group interchange transform. It works like GP1, but it is called only when the user selects auxiliary mode.
- FGI2 - two-group functional group interchange transform. It works like GP2, but it is called only when the user selects aux. mode.


```

.rxn
name "transform name"
type GP0
[rating = rating]
[.conditions]
[.comments]
"comment"
"comment"
..]

.start
[statement
statement
statement
...]
```

Figure 6 GP0 transform structure

```

.rxn
name "transform name"
type RING
RINGTYPE ring type list
[rating = rating]
[.conditions]
[.comments]
"comment"
"comment"
..]

.start
[statement
statement
statement
...]
```

Figure 7 RING transform structure

- FGA - functional group addition. It has no retron; there is only a user-specified atom which is assigned to variable A1.
- GP0 - zero-group transform. There is no retron. These transforms may be applied to a particular bond if selected by the user.

To add new types it is necessary to modify the source code of the transform interpreter.

G1 functional group list

Specifies the functional group that keys a GP1 or FGI1 transform or one of the two groups for a GP2 or FGI2 transform. It is a comma-separated list of functional group names. The names must match those in the file funcgrp.chm.

[G2 functional group list]

Used only in GP2 and FGI2 transforms; specifies the second group that keys the transform.

Note that if GP1 = GP2, the transform is called twice. When the effect of the transform is symmetrical this is redundant (for example, the ozonolysis transform of a 1,6-diketone); but when the effect of the transform is asymmetrical it is really necessary to call it twice (for example, the acylation transform of a 1,3-diketone).

```

.rxn
name "transform name"
type FGI1
gprec functional group list
[rating =rating]
[link "url"]
[.conditions]
[.comments]
"comment"
"comment"
...}

.start
[statement
statement
statement
...}

```

Figure 8 FGA transform structure

[path *pathlength*]

Used only in GP2 and FGI2, but in such transforms it is a required field. The parameter *pathlength* is an integer. The length is the number of atoms between the root atoms of the two groups, *including* both root atoms. Therefore the minimum length is usually two.

[gprec=*functional group list*]

Used in FGI1, FGI2, and FGA transforms. It indicates the functional group that is left in the precursor after the execution of the transform.

[rating=*rating*]

A priori rating. If not specified, the default is 50. *Rating* is a predefined numerical variable and may be modified in the transform body if necessary.

[link "*url*"]

A link to a webpage with further information about the transform. The *url* may be relative or absolute. The link is opened in a new browser window when the user clicks the *more info* button.

[.conditions]

Conditions block. At the present time it is ignored by the interpreter. The block ends at the beginning of the next block (usually .comments or .start).

[.comments]

Comments block. The comments are a list of double quote-enclosed strings. The strings are automatically concatenated. Newline characters may be specified with the \n (backslash-en) sequence within the string. The comments block ends at the next header field.

.start

Marks the end of the header and the beginning of the body.

Statements

The body of the transform is a statement list and goes after .start. There are several types of statements: assignment, branching, atom and bond manipulation, output, and done.

Assignment

variable = numerical expression

The following is an example of a valid assignment:

```
rating = rating + 10
```

Branching

```
if (Boolean expression) then
```

```
    statement list (true)
```

```
[else
```

```
    statement list (false)]
```

```
endif
```

If the expression is true, the *statement list (true)* is executed. If it is false, the *statement list (false)* is executed. The later statement list is optional. The `endif` is required, even if the statement list is null or just one line.

foratom (*atom1* from *atom0*)

statement list

next

This is used to visit all of the neighbors of the atom specified by *atom0*. The current neighbor is stored in the variable *atom1*. *Atom0* is not affected, but *atom1* takes a new value for each iteration. *Atom0* and *atom1* are atom-type variables.

Atom and bond manipulation

breakbond(*atom1*, *atom2*)

Breaks the bond between the specified atoms. If the bond is double or triple, it breaks it completely.

makebond(*atom1*, *atom2*)

Creates a single bond between *atom1* and *atom2*. If they are already bonded, the bond order is incremented.

incbondorder (*atom1*, *atom2*)

Increments the bond order between *atom1* and *atom2*. This works only if the atoms are already bonded. Notice that, for bonded atoms, *incbondorder* and *makebond* do the same thing.

decbondorder(*atom1*, *atom2*)

Decreases the order of the bond between *atom1* and *atom2*. If it is a single bond, the bond is broken.

setbondorder(*atom1*, *atom2*, *order*)

Sets the order of the bond between *atom1* and *atom2*. The bond must exist. *Order* must be an integer constant or variable. If *order* is less than or equal to zero, the bond is broken; if it is impossibly high, the bond is set at the maximum possible order.

`add(atom I, element)`

Add creates a new atom of the specified element and joins it to *atom I* with a single bond. *Element* must be a conventional atomic symbol (but it is not case-sensitive). Add may be used as a function; see page 108. Example:

```
add(a1, O)
```

`done`

`done (atom list)`

This is an extremely important statement and can be easily overlooked. Do not forget to put at least one `done` statement in each transform. After all operations required by the transform are applied to the molecule, the `done` statement is used to save the result. If there is no `done` statement, the transform is executed but it returns no results!

The execution continues at the next statement after `done`, with a new copy of the molecule as it was before the transform execution started. However, the rating is not reset. In some cases it may be necessary to reset the rating after the `done` statement.

If `done` is within a loop, several different molecules may be generated by a single execution of a transform. There may also be more than one `done` statement in a transform.

`Done` may optionally receive a comma-separated atom list as a parameter. All the molecules which include the listed atoms are deleted. This is often used to delete inorganic or standard reagents such as water or ammonia.

Output

`echo("string")`

Sends the *string* to the client. It is shown along with the standard comments about the reaction. This is used typically to explain rating changes or other structural conditions that could affect the transform.

methyl	primary	secondary
tertiary	quaternary	vinyl
carbonyl	alkynyl	nitrile
allene	alkyl	sp3
sp2	sp	ether
nitrile	hydroxyl	nitro
peroxide	carbonyl	benzyl
allyl	alpha_nitrile	alpha_carbonyl
alpha_alkynyl	alpha_CH	phenyl
EWG	alpha_EWG	alpha_EWG2
alpha_EWG3	C	N
H	O	S
X	Cl	Br
I		

Figure 9 Atom types.

Functions

`ringsize(atom1, atom2)`

Returns an integer. The return value is the size of the smallest ring containing both atoms.

If the atoms are not in a ring, the function returns zero.

`add(atom1, element)`

Returns an atom. Add creates a new atom of the specified element and joins it to *atom1* with a single bond. *Element* must be a conventional atomic symbol (but it is not case-sensitive). Example:

```
my_new_oxygen = add(a1, O)
```

Add may be used as a statement; in that case the return value is not used.

Expressions and Operators

Expressions are very similar to those in other languages such as Pascal or BASIC; they may contain nested parentheses to specify the desired priority. There are atom type operators, Boolean operators, numerical comparison operators, and math operators.

Atom-type operators

`atom is atom_type`

`atom isnot atom_type`

These are used in expressions to check whether a given atom is of a given type. The types are reserved words, and are listed in Figure 9. The `is / isnot` operator has lower priority than the Boolean operators. **Note:** The types are assigned before the execution of the transform and are not updated during execution.

atom iseq atom

This compares two atoms topologically, returning true if they are equivalent. For example, the two methyl groups in acetone would be topologically equivalent. This applies to the atoms as they were before the transform started; for example, if one of the methyl groups of acetone were changed, this operator would still consider it equivalent to the other group.

Numerical comparison operators

`>, >=, <, <=, <>, =`

Notice that the equality operator is the same as the assignment operator (surely this will be disliked by mathematicians and philosophers). The equality and not-equals operator may be used with variables of any type; the other comparison operators are only valid for numerical types.

Boolean (logical) operators

`and, or, not`

The name says it all. The priority follows the typical order: `not > and > or`.

Math operators

`+, -, *, /`

As usual, multiplication and division have a higher priority than addition and subtraction. Associativity is from left to right.

Types and variables

The available variable types are numerical (integer) and atom. Variables do not need to be declared; their type is automatically assigned by the program. However, the typing is

global. That means that once a variable type is assigned the same variable cannot be used with a different type, even in another transform. Other internal variable types include string and functional groups, but currently they cannot be manipulated, so they are more like constants.

There are several pre-defined variables.

A1–A9, B1–B9, P1–P9, R1–R9, S1–S9

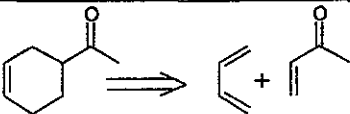
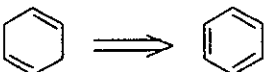

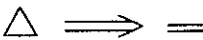
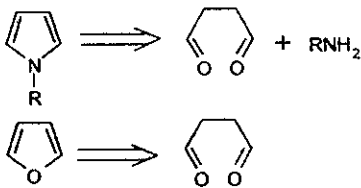
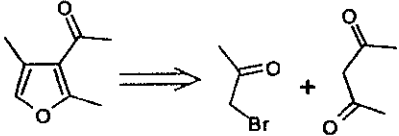
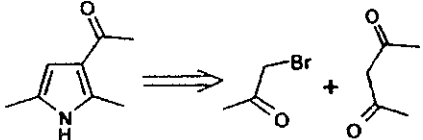
These are all atom-type variables. They are assigned automatically at the beginning of each transform and are used as the basis for all atom operations. A_N are the atoms of the first functional group, and are assigned according to the definition of the functional group (see the functional group description language guide). B_N are likewise defined for the second functional group; obviously they are only useful in GP2-type transforms. P_N are defined for the path between the two groups: P_1 is the root of the first group, and the numbering goes on until the root of the second group. Notice that, for a 2-group transform, several atoms are assigned more than one variable (for example, A_i is always the same as P_i). R_N , S_N , etc. variables are defined for ring transforms. R_1 to R_N are the atoms of the first ring, numbered according to the pattern which describes the ring. S_1 to S_N are the atoms of the second ring in a fused ring system.

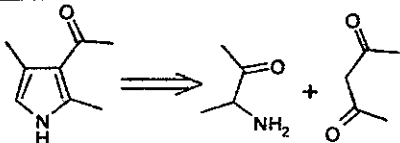
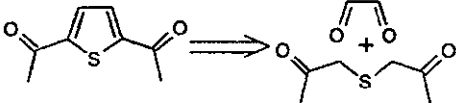
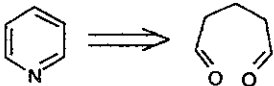
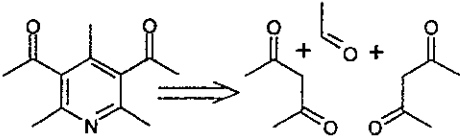
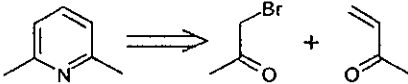
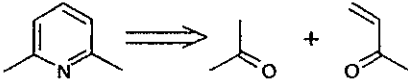
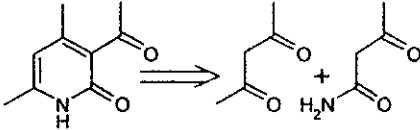
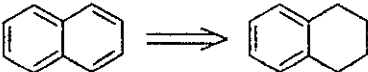
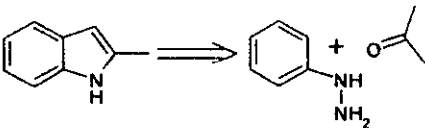

Rating

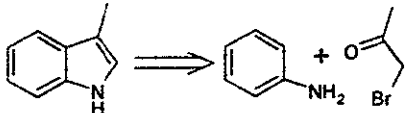
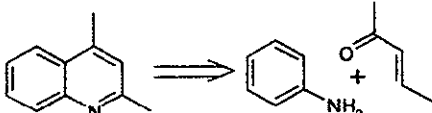
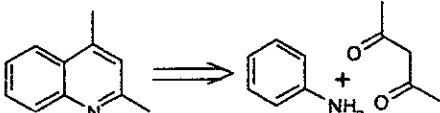
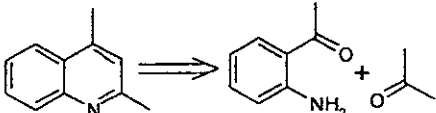
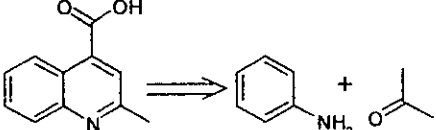
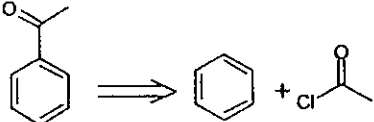
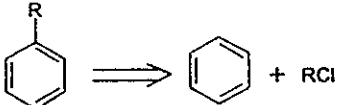


`rating` is a numerical variable used to evaluate how chemically useful a transform is. It is a semi-quantitative measure of the yield, generality, cost, etc. of the reaction. It should be modified as necessary by the transform body if structural conditions warrant it.

C. ÍNDICE DE TRANSFORMADAS

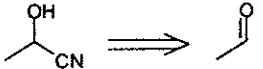

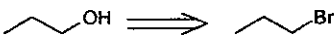
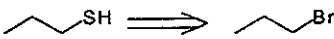
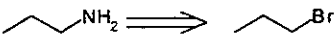
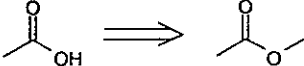
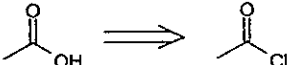
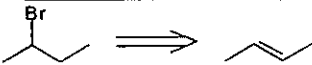

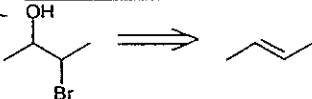
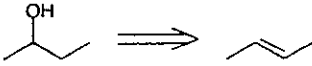
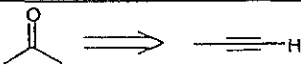
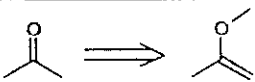
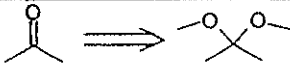
La lista mostrada a continuación contiene todas las transformadas incluidas en la base de datos de OSET al momento de redactar esta tesis.

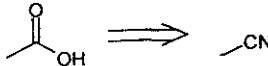
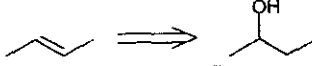
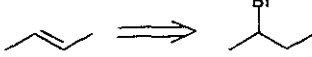
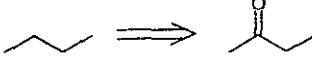
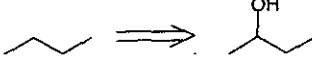
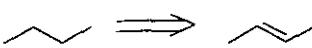
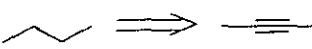
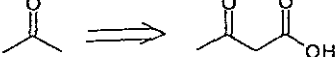

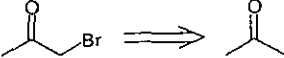
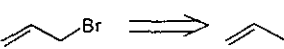
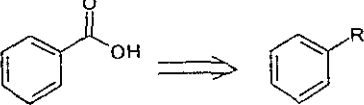
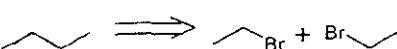
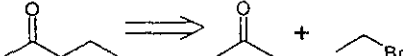
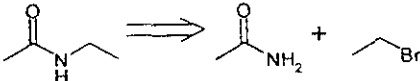
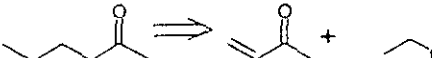
Nombre	Ejemplo
1. Reacción de Diels-Alder	
2. Reducción de Birch	
3. Cicloadición 2 + 2	
4. Adición de carbeno a doble enlace C-C	
5. Síntesis de furanos y pirroles de Paal-Knorr	
6. Síntesis de furanos de Feist-Benary	
7. Síntesis de pirroles de Hantzsch	

Nombre	Ejemplo
8. Síntesis de pirroles de Knorr	
9. Síntesis de tiofenos de Hinsberg	
10. Síntesis de piridinas de Knövenagel	
11. Síntesis de piridinas de Hantzsch	
12. Síntesis de piridinas de Kröhnke	
13. Síntesis de piridinas de Corey	
14. Síntesis de 2-piridona de Guareschi-Thorpe	
15. Deshidrogenación del 1,2,3,4 - tetrahidronaftaleno	
16. Síntesis de indoles de Fischer	
17. Síntesis de indoles de Madelung	

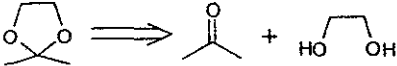
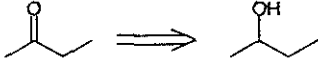
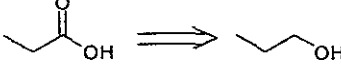
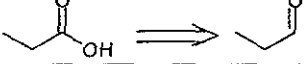
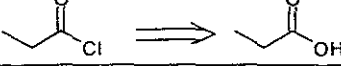
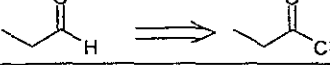
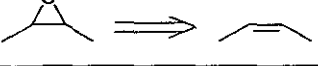
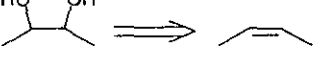
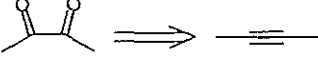
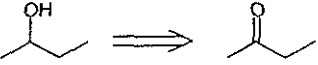
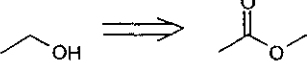
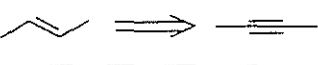
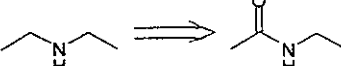
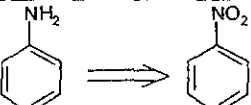
Nombre	Ejemplo
18. Síntesis de indoles de Bischler	
19. Síntesis de quinolinas de Skraup	
20. Síntesis de quinolinas de Combes	
21. Síntesis de quinolinas de Friedlander	
22. Síntesis de quinolinas de Pfitzinger	
23. Acilación de Friedel-Crafts	
24. Alquilación de Friedel-Crafts	
25. Halogenación aromática	
26. Nitración aromática	

Nombre	Ejemplo
27. Condensación aldólica	
28. Condensación de Claisen o de Dieckmann	
29. Acilación de cetonas, nitrilos o ácidos carboxílicos con ésteres carboxílicos	
30. Adición de Michael de enolatos	
31. Adición de Michael de hetero nucleófilos	
32. Condensación aldólica/deshidratación	
33. Reacción de Mannich	
34. Ozonólisis reductiva	
35. Ozonólisis oxidativa	
36. Hidrólisis de furanos	
37. Condensación benzoinica	
38. Transposición de Claisen	

Nombre	Ejemplo
39. Formación de cianhidras	 <chem>CC(O)C + CN#N >> CC(O)C#N</chem>
40. Halogenación de alcoholes	 <chem>CCCO + PBr3 >> CCB + POBr2</chem>
41. Hidrólisis de bromuros de alquilo	 <chem>CCCB + OH- >> CCCO + Br-</chem>
42. Tiólisis de bromuros de alquilo	 <chem>CCCB + S- >> CCCS + Br-</chem>
43. Alquilación del amoniaco	 <chem>CCCB + NH3 >> CCNC + Br-</chem>
44. Hidrólisis de ésteres	 <chem>CC(=O)OC + H2O >> CC(=O)O + CH3OH</chem>
45. Hidrólisis de cloruros de acilo	 <chem>CC(=O)Cl + H2O >> CC(=O)O + HCl</chem>
46. Adición de HBr a enlaces dobles	 <chem>CCC=CC + HBr >> CCC(Br)CC</chem>
47. Adición de bromo a enlaces dobles	 <chem>CCC=CC + Br2 >> CCC(Br)C(Br)CC</chem>
48. Adición de HOBr a enlaces dobles	 <chem>CCC=CC + HOBr >> CCC(O)C(Br)CC</chem>
49. Hidratación de alquenos	 <chem>CCC=CC + H2O >> CCC(O)CC</chem>
50. Hidratación de alquinos	 <chem>C#C + H2O >> CC=O</chem>
51. Hidrólisis de éteres enólicos	 <chem>CC(OC)=C + H2O >> CC(=O)C + CH3OH</chem>
52. Hidrólisis de acetales	 <chem>CC(OC)(OC)C + H2O >> CC(=O)C + 2 CH3OH</chem>

Nombre	Ejemplo
53. Hidrólisis de nitrilos a ácidos carboxílicos	
54. Deshidratación de alcoholes	
55. Deshidrohalogenación	
56. Reducción de Clemmensen o Wolff-Kishner	
57. Reducción de alcoholes	
58. Hidrogenación catalítica de olefinas	
59. Hidrogenación catalítica de alquinos	
60. Descarboxilación	
61. Hidrólisis de éster y descarboxilación	
62. Bromación alfa a carbonilo	
63. Bromación alílica	
64. Oxidación de cadena aromática	
65. Acoplamiento de halogenuros de alquilo con reactivo organometálico	
66. Alquilación de enolatos	
67. N-Alquilación de amidas	
68. Adición de Michael de organocupratos	

Nombre	Ejemplo
69. Esterificación	<chem>CC(=O)OCC >> CC(=O)O + OCC</chem>
70. Amidación	<chem>CC(=O)NCC >> CC(=O)O + NCC</chem>
71. Transposición de Favorskii	<chem>C1CCC1C(=O)O >> O=C1CCCC1Br</chem>
72. Adición de Grignard a carbonilo	<chem>CCC(O)C >> CCC(=O)C + BrCC</chem>
73. Adición de Grignard al dióxido de carbono	<chem>CCC(=O)O >> BrCC</chem>
74. Adición de reactivo organometálico a epóxido	<chem>CCOC1CO1 >> BrCC + OCC</chem>
75. Formación de nitrilos a partir de halogenuros de alquilo	<chem>BrCC >> CCC#N</chem>
76. Reacción de Wittig	<chem>BrCC >> CCC=O</chem>
77. Adición de Grignard a nitrilo	<chem>BrCC + CCC#N >> CCC(=O)C</chem>
78. Adición de Grignard a éster	<chem>BrCC + CCC(=O)OC >> CCC(=O)CC</chem>
79. Síntesis de éteres de Williamson	<chem>BrCC + OCC >> COCC</chem>
80. Alquilación de aminas	<chem>BrCC + NCC >> CCNCC</chem>
81. Alquilación de alquinos terminales	<chem>BrCC + C#CC >> C#CCC</chem>
82. Aminación reductiva	<chem>CCC(=O)C + RNHR >> CCC(N)C + R2O</chem>

Nombre	Ejemplo
83. Formación de acetales	 <chem>CC(=O)C.OCCO>>CC(OC1OCCO1)C</chem>
84. Oxidación de alcoholes a cetonas o aldehídos	 <chem>CCO>>CC=O</chem>
85. Oxidación de alcoholes a ácidos carboxílicos	 <chem>CCO>>CC(=O)O</chem>
86. Oxidación de aldehídos a ácidos carboxílicos	 <chem>CC=O>>CC(=O)O</chem>
87. Formación de cloruros de acilo	 <chem>CC(=O)O>>CC(=O)Cl</chem>
88. Reducción de cloruros de acilo	 <chem>CC(=O)Cl>>CC=O</chem>
89. Oxidación de alquenos a oxiranos	 <chem>C=C>>C1CO1</chem>
90. Dihidroxilación de alquenos	 <chem>C=C>>C(O)CO</chem>
91. Oxidación de alquinos a dicetonas	 <chem>C#C>>CC(=O)C(=O)C</chem>
92. Reducción de carbonilos	 <chem>CC=O>>CCO</chem>
93. Reducción de ésteres a alcoholes	 <chem>CC(=O)OC>>CCO</chem>
94. Hidrogenación parcial de alquinos	 <chem>C#C>>C=C</chem>
95. Reducción de amidas	 <chem>CC(=O)N>>CCO</chem>
96. Reducción de grupo nitro	 <chem>c1ccc(cc1)[N+](=O)[O-]>>c1ccc(cc1)N</chem>

REFERENCIAS

1. AGARWAL, K. K. (1998) "An Algorithm for Computing the Automorphism Group of Organic Structures with Stereochemistry and a Measure of its Efficiency". *J. Chem. Inf. Comput. Sci.* **38**, 402-404, 1998.
2. AYYAR, S., K.; RAO, G. S. K. (1968) "Studies in Terpenoids. IV. Synthetic Studies in Juvabiones and Analogues. Conversion of ar-(+)-turmerone to ar-(+)-juvabione". *Canad. J. Chem.* **46**, 1467-1472, 1968.
3. BARONE, R.; RIBERO, B.; GIBERT, B.; MEYER, R. (1985) "Microcomputer-Assisted Drills in Organic Synthesis". *J. Chem. Educ.* **62**[5], 411, 1985.
4. BARONE, R.; CHANON, M. (1986) "Computer-Aided Organic Synthesis (CAOS)" en VERNIN, G.; CHANON, M. (eds.) *Computer Aids to Chemistry*. Ellis Horwood, 1986, pp. 19-102.
5. BARONE, R.; ARBELOT; CHANON, M. (1995) "CONAN (CONnectivity ANalysis): A Simple Approach in the Field of Computer-Aided Organic Synthesis. Example of the Taxane Framework". *J. Chem. Inf. Comput. Sci.* **35**, 467-471, 1995.
6. BAUMER; COMPAGNARI; SALA; SELLO (1992) "Erythronolide B: Analysis of the Synthesis Pathway Proposed by the Lilith Program". *Recl. Trav. Chim. Pays-Bas* **111**, 297-303, 1992.
7. BENECKE, C.; GRUND, R.; HOHNBERGER, R.; KERBER, A.; LAUE, R.; WIELAND, T. (1995) "MOLGEN+, a generator of connectivity isomers and stereoisomers for molecular structure elucidation". *Analytica Chimica Acta* **314**, 141-147, 1995.
8. BENECKE, C.; GRUND, R.; KERBER, A.; LAUE, R.; WIELAND, T. (1995) "Chemical Education via MOLGEN". *J. Chem. Educ.* **72**[5], 403, 1995.
9. BERGNER, E. J.; HELMCHER, G. (2000) "Enantioselective Synthesis of (+)-Juvabione". *J. Org. Chem.* **65**, 5072-5074, 2000.
10. BERSOHN, M.; ESACK, A. (1976) "Computers and Organic Synthesis". *Chemical Reviews*, **76**, 269-282, 1976.
11. BERTRAND, M. P.; MONTI, H.; BARONE, R. (1986) "Computer-Assisted Organic Synthesis: An Undergraduate Experiment". *J. Chem. Educ.*, **63**, 624, 1986.
12. BOITEN, J.; NOORDIK, J. H.; GROEN, M. B. (1993) "Cationic Polyene Cyclizations. A Computer Assisted Synthesis Approach". *J. Chem. Inf. Comput. Sci.* **33**, 727-735, 1993.
13. BONE, R. G. A.; FIRTH, M. A.; SYKES, R. A. (1999) "SMILES Extensions for Pattern Matching and Molecular Transformations. Applications in Chemoinformatics". *J. Chem. Inf. Comput. Sci.* **39**, 846-860, 1999.
14. BRANDT, J. et al. (1977) "Computer Programs for the Deductive Solution of Chemical Problems on the Basis of a Mathematical Model of Chemistry" en WIPKE, W.T.; HOWE, W. J. (eds.) *Computer-Assisted Organic Synthesis*. Washington: ACS, 1977.
15. BROWN, H. C. (1959) "Foundations of the Structural Theory". *J. Chem. Ed.* **36**:3, 104-110, 1959
16. BUSH, B. L.; SHERIDAN, R. P. (1993) "PATTY: A Programmable Atom Typer and Language for Automatic Classification of Atoms in Molecular Databases". *J. Chem. Inf. Comput. Sci.* **33**, 756-762, 1993.
17. CABROL, D.; CACHET, C.; CORNELIUS, R. (1986) "Teaching Chemistry with Microcomputers". en Vernin, Chanon (eds), *Computer Aids to Chemistry*, Ellis Horwood Ltd, 1986.

18. CAPRON; PERRON (1993) *Computers & Information Systems: Tools for an Information Age*. Benjamin / Cummings, 1993.
19. CAREY; SUNDBERG (1990a) *Advanced Organic Chemistry, 3rd ed. part B: Reactions and Synthesis*. Plenum, 1990.
20. CAREY; SUNDBERG (1990b) *Advanced Organic Chemistry, 3rd ed. part A: Structure and Mechanism*. Plenum, 1990.
21. CARTWRIGHT (1993) *Applications of Artificial Intelligence in Chemistry*. Oxford Chemistry Primers, Oxford University Press, 1993.
22. CHODOSH; HILL; SHPILSKY; MENDELSON (1992) "SYNthesis LIBrary, an Expert System for Chemical-reaction Knowledge-base Management". *Recl. Trav. Chim. Pays-Bas*, **111**, 247-254, 1992.
23. CHOPLIN, F.; GOUNDIAM, S.; KAUFMANN, G. (1983) "Storage and Retrieval of Synthetic Trees". *J. Chem. Inf. Comput. Sci.* **23**, 26-30, 1983.
24. COREY, E. (1967) *Pure and Applied Chem.*, **14**, 19-37, 1967.
25. COREY, E.; WIPKE, W. T. (1969) "Computer-Assisted Design of Complex Organic Syntheses". *Science*, **166**, 178-192, 1969.
26. COREY, E. (1971) *Quart. Rev. Chem. Soc.*, **25**, 455-482, 1971.
27. COREY, E. J. *et al.* (1980) "Computer-Assisted Synthetic Analysis. Long-Range Search Procedures for Antithetic Simplification of Complex Targets by Application of the Halolactonization Transform". *J. Chem. Inf. Comput. Sci.*, **20**, 221-230, 1980.
28. COREY, E. J.; LONG, A.K.; RUBENSTEIN, S.D. (1985a) "Computer-Assisted Analysis in Organic Synthesis". *Science*, **228**, 408-418, 1985.
29. COREY, E. J. *et al.* (1985b) "Computer Assisted Synthetic Analysis. Selection of Protective Groups for Multistep Organic Syntheses". *J. Org. Chem.*, **50**, 1920-1927, 1985.
30. COREY, E. J.; CHENG, X. (1989) *The Logic of Chemical Synthesis*. New York: Wiley, 1989.
31. COREY, E. J.; LONG, A. K.; LOTTO, G. I.; RUBENSTEIN, S. D. (1992) "Computer-assisted Organic Synthesis. Quantitative Assesment of Transform Utilities". *Recl. Trav. Chim. Pays-Bas*, **111**, 304-309, 1992.
32. CSIZMADIA, F. (2000) "JChem: Java Applets and Modules Supporting Chemical Database Handling from Web Browsers". *J. Chem. Inf. Comput. Sci.* **40**, 323-324, 2000.
33. DALBY, A. *et al.* (1992) "Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited". *J. Chem. Educ. Inf. Comput. Sci.*, **32**, 244-255, 1992.
34. DENGLE, A.; FONTAIN, E.; KNAUER, M.; STEIN, N.; UGI, I. (1992) "Competing Concepts in CAOS". *Recl. Trav. Chim. Pays-Bas*, **111**, 262-269, 1992.
35. DITTMAR, P. G.; MOCKUS, J.; COUVREUR, K. M. (1977) "An Algorithmic Computer Graphics Programs for Generating Chemical Structures". *J. Chem. Inf. Comput. Sci.* **17**, 186-192, 1977.
36. DOENGES, R.; GROEBEL, B.; NICKELSEN, H.; SANDER, J. (1985) "TOSCA: A Topological Synthesis Design by Computer Application". *J. Chem. Inf. Comput. Sci.* **25**, 425-430, 1985.
37. DOGANE; TAKABATAKE, T.; BERSOHN, M. (1992) "Computer-executed synthesis planning, a progress report". *Recl. Trav. Chim. Pays-Bas*, **111**, 291-296, 1992
38. DOWNS, G. M.; GILLET, V. J.; HOLLIDAY, J. D.; LYNCH, M. F. (1989) "Review of Ring Perception Algorithms for Chemical Graphs". *J. Chem. Inf. Comput. Sci.* **29**, 172-187, 1989.
39. DUGUNDJ, J.; GILLESPIE, P.; MARQUARDING, D.; UGI, I.; RAMÍREZ, F. (1976) "Metric Spaces and Graphs Representing the Logical Structure of Chemistry". en Balaban, A. T., *Chemical Applications of Graph Theory*, Academic Press, 1976.
40. ELK, S. B. (1996) "Orismology (the Science of Defining Words) and the Geometrical Foundations of Chemistry. 4. Polycyclic Compounds". *J. Chem. Inf. Comput. Sci.* **36**, 385-392, 1996.

41. FAN, B. T.; PANAYE, A.; DOUCET, J.; BARBU, A. (1993) "Ring Perception. A New Algorithm for Directly Finding the Smallest Set of Smallest Rings form a Connection Table". *J. Chem. Inf. Comput. Sci.* **33**, 657-662, 1993.
42. FELDMAN, M.; BISHOP, M. (1976) "Computer-Assisted Instruction in Organic Synthesis". *J. Chem. Educ.* **53**[2], 91, 1976.
43. FIGUERAS, J. (1993) "Morgan Revisited". *J. Chem. Inf. Comput. Sci.* **33**, 717-718, 1993.
44. FIGUERAS, J. (1996) "Ring Perception Using Breadth-First Search". *J. Chem. Inf. Comput. Sci.*, **36**, 986-991, 1996.
45. FREGGA, A. The Electronic Nobel Museum Project: Nobel Prize in Chemistry 1965. <http://www.nobel.se/laureates/chemistry-1965-press.html> (consultado en mar/00).
46. FUJITA, S. (1988) "A New Algorithm for Selection of Synthetically Important Rings. The Essential Set of Essential Rings for Organic Structures". *J. Chem. Inf. Comput. Sci.* **28**, 78-82, 1988.
47. GASTEIGER, J.; JOCHUM, C. (1979) "An Algorithm for the Perception of Synthetically Important Rings". *J. Chem. Inf. Comput. Sci.* **19**, 43-48, 1979.
48. GASTEIGER, J.; IHLENFELDT, W.; RÖSE (1992) "A collection of computer methods for synthesis design and reaction prediction". *Recl. Trav. Chim. Pays-Bas* **111**, 270-290, 1992.
49. GELERNTER, H. L. *et al.* (1977) "Empirical Explorations of SYNCHÉM". *Science*, **197**, 1041-1049, 1977.
50. GHOSHAL, N. (1990) "Yet another Linear Notation Scheme for Organic Compounds. 1". *J. Chem. Inf. Comput. Sci.* **30**, 308-311, 1990.
51. GILCHRIST (1992) *Química Heterocíclica*, 2a edición. Addison-Wesley Iberoamericana, 1995.
52. GONDRAN M.; MINOUX M. (1979) *Graphs and Algorithms*. John Wiley and Sons, 1990
53. GUND, P.; GRABOWSKI, E. J. J.; HOFF, D. R.; SMITH, G. M.; ANDOSE, J. D.; RHODES, J. B., WIPKE, W. T. (1980) "Computer-Assisted Synthetic Analysis at Merck". *J. Chem. Inf. Comput. Sci.* **20**, 88-93, 1980.
54. HAGGIN, J. (1983) "Computers Shift Chemistry To More Mathematical Basis". *C&EN*, may 9, 1983, pp. 7-20.
55. HAMM; JAUFFRET; KAUFMANN (1992) "Generalization and representation of synthesis pathways in the COSYMA system". *Recl. Trav. Chim. Pays-Bas*, **111**, 317-322, 1992.
56. HANSEN, P. J.; JURIS, P. C. (1988) "Chemical Applications of Graph Theory". *J. Chem. Educ.* **65**[7], 574, 1988.
57. HANSER, T.; JAUFFRET, P.; KAUFMANN, G. (1996) "A New Algorithm for Exhaustive Ring Perception in a Molecular Graph". *J. Chem. Inf. Comput. Sci.* **36**, 1146-1152, 1996.
58. HELSON, H. E. (1999) "Structure Diagram Generation". en Lipkowitz, K.B. y Boyd, D. B (eds.) *Reviews in Computational Chemistry*, Vol. 13, Wiley, NY, 1999.
59. HENDRICKSON, J. B.; HUANG, P.; TOCZKO, G. (1987) "Molecular Complexity: A Simplified Formula Adapted to Individual Atoms". *J. Chem. Inf. Comput. Sci.* **27**, 63-67, 1987.
60. HENDRICKSON, J. B. *et al.* (1989a) "New Directions in the SYNGEN Program for Synthesis Design", en Hohne, B.A.; Pierce, T.H. (eds.) *Expert System Applications in Chemistry*. Washington, ACS, 1989.
61. HENDRICKSON, J. B.; HUANG (1989b) "Multiple Constructions in Synthesis Design". *J. Chem. Inf. Comput. Sci.* **29**, 145-151, 1989.
62. HENDRICKSON, J. B., TOCZKO, G. (1989c) "SYNGEN Program for Synthesis Design. Basic Computing Techniques". *J. Chem. Inf. Comput. Sci.* **29**, 137-145, 1989.
63. HENDRICKSON, J. "Organic Synthesis in the Age of Computers". *Angew. Chem. Int. Ed. Engl.*, **29**, 1286-1295 (1990).
64. HENDRICKSON, J. B. (1992a) "Descriptions of reactions: their logic and applications" *Recl Trav. Chim. Pays-Bas*, **111**, 323-334, 1992

65. HENDRICKSON, J. B.; PARKS (1992b) "A program for the Forward Generation of Synthetic Routes". *J. Chem. Inf. Comput. Sci.* **32**, 209-215, 1992.
66. HENDRICKSON, J. B. (1998) "Building the Shortest Synthesis Route". *CHEMTECH* **28**(9), 35-40, 1998.
67. HIPPE, Z. (1981) "Self-Adapting Computer Program System for Designing Organic Syntheses". *Analytica Chimica Acta*, **133**, 677-683, 1981.
68. HIPPE; FIC; MAZUR (1992) "A preliminary appraisal of selected problems in computer-assisted organic synthesis". *Recl. Trav. Chim. Pays-Bas*, **111** 255-261, 1992.
69. HOFFMANN, R. (1995) *Lo mismo y no lo mismo*. Fondo de Cultura Económica, México 1997.
70. HOPF, H.; MAAS, G. (1992) "Preparation and Properties, Reactions and Applications of Radialenes". *Angew. Chem. Int. Ed. Engl.*, **31**[4], 931-54, 1992.
71. HOUSE, H. O. (1972) *Modern Synthetic Reactions*, second edition. W. A. Benjamin Inc., 1972.
72. HU, C.; XU, L. (1996) "On Highly Discriminating Molecular Topological Index". *J. Chem. Inf. Comput. Sci.* **36**, 82-90, 1996.
73. IHLENFELDT, W. (1994) "An Efficient Approach toward a Flexible and General Knowledge Definition and Program Control Language for a Synthesis-Planning Program". *J. Chem. Inf. Comput. Sci.* **34**, 872-880, 1994.
74. IHLENFELDT, W.; GASTEIGER, J. (1995) "Computer-Assited Planning of Organic Syntheses: The Second Generation of Programs". *Angew. Chem. Int. Ed. Engl.*, **34**, 2613-2633, 1995.
75. JOHNSON, P. Y. *et al.* (1989) "Designing an Expert System for Organic Synthesis: The Needs for Strategic Planning", en HOHNE, B.A.; PIERCE, T.H. (eds.) *Expert System Applications in Chemistry*. Washington, ACS, 1989.
76. JOHNSON, A. P.; MARSHALL, C.; JUDSON, P. N. (1992) "Some recent progress in the development of the LHASA computer system for organic synthesis design: Starting-material-oriented retrosynthetic analysis". *Recl. Trav. Chim Pays-Bas*, **111**, 310-316, 1992.
77. JORGENSEN, W. L.; LAIRD; GUSHURST; FLEISCHER; GOTHE; HELSON; PADERES; SINCLAIR (1990) "CAMEO: a program for the logical prediction of the products of organic reactions". *Pure & Appl. Chem.* **62**, 1921-1932, 1990.
78. KREBSBACH, D.; GELERNTER, H.; SIEBURTH, S. (1998) "Distributed Heuristic Synthesis Search". *J. Chem. Inf. Comput. Sci.*, **38**, 595-604, 1998.
79. LEWELL; JUDD; WATSON; HANN (1998) "RECAP-Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry". *J. Chem. Inf. Comput. Sci.* **38**, 511-522, 1998.
80. LONG, A. K.; RUBENSTEIN, S. D.; JONCAS, L. J. (1983) "A Computer Program for Organic Synthesis". *C&EN*, may 9, 1983, pp. 22-30.
81. LUSHNIKOV; ZEFIROV (1992) "Fragmentations and the 'Feeling of Goal' in Computer-Assisted Synthesis". *J. Chem. Inf. Comput. Sci.* **32**, 317-322, 1992.
82. MAK (1991) *Writing Compilers and Interpreters: An Applied Approach*. John Wiley and Sons, 1991.
83. MARCH, J. (1992) *Advanced Organic Chemistry: Reactions, Mechanisms and Structure*, 4th ed. Wiley-Interscience, 1992.
84. MATYSKA, KOCA (1991) "MAPOS: A Computer Program for Organic Synthesis Design Based on Synthon Model of Organic Chemistry" *J. Chem. Inf. Comput. Sci.* **31**, 380-386, 1991.
85. MITCHELL, D. K.; CHAMBRON, J (1995) "Chemical Topology. The Ins and Outs of Molecular Structure". *J. Chem. Educ.* **72**[12], 1059, 1995.
86. MORGAN, H. L. (1965) "The Generation of a Unique Machine Description for Chemical Structures—A Technique Developed at Chemical Abstracts Service" *J. Chem. Doc.* **5**, 107-113, 1965.

87. MORI, K.; MATSUI, M. (1967) "Synthesis of Compounds with Juvenile Hormone Activity—(±)-Juvanione (Methyl (±)-Todomatuate)". *Tetrahedron* **24**, 3127-3138, 1967.
88. NAKAYAMA, T. (1994) "Semantic Dictionary for Substructure Matching of Chemical Structures with General Descriptors". *J. Chem. Inf. Comput. Sci.* **34**, 845-853, 1994.
89. NILAKANTAN, R.; BAUMAN, N.; HARAKI, K. S.; VENKATARAGHAVAN, R. (1990) "A Ring-Based Chemical Structure Query System: Use of a Novel Ring Complexity Heuristic". *J. Chem. Inf. Comput. Sci.* **30**, 65-68, 1990.
90. ORF, H. W. (1975) "Computer-Assisted Instruction in Organic Synthesis". *J. Chem. Educ.*, **52**, 464-467, 1975.
91. OTT, M. A.; NOORDIK, J. H. (1992) "Computer Tools for Reaction Retrieval and Synthesis Planning in Organic Chemistry". *Recl. Trav. Chim. Pays-Bas*, **111**, 239-246, 1992.
92. OTT, M. A.; NOORDIK, J. H. (1997) "Long-Range Strategies in the LHASA Program: The Quinone Diels-Alder Transform". *J. Chem. Inf. Comput. Sci.*, **37**, 98-108, 1997.
93. OTT, M. A. Overview of the LHASA System. <http://www.cmbi.kun.nl/cheminf/olp/> (consultado en mar/00).
94. OTT, M. A. Analysis of Complex Synthetic Problems. <http://www.cmbi.kun.nl/cheminf/olp/> (consultado en mar/00).
95. OUYANG, Z.; YUAN, S.; BRANDT, J.; ZHENG, C. (1999) "An Effective Topological Symmetry Perception and Unique Numbering Algorithm". *J. Chem. Inf. Comput. Sci.* **39**, 299-303, 1999.
96. PAWSON, B. A.; CHEUNG, H. C.; GURBAXANI, S.; SAUCY, G. (1970) "Syntheses of Natural (+)-Juvabione, Its Enantiomer (-)-Juvabione, and Their Diastereoisomers (+)- and (-)-Epjubavione". *JACS* **92**:2, 336-343, 1970.
97. PENSAK; COREY, E. J. (1977) "LHASA-Logic and Heuristics Applied to Synthetic Analysis" en WIPKEY HOWE(eds), *Computer Assisted Organic Synthesis, ACS Symposium Series* **61**, 1977, p. 1.
98. POLYA, G. (1944) *Cómo plantear y resolver problemas*, Trillas, México 1965 (reimp. 1999)
99. POZHARSKII, A. F.; SOLDATENKOV, A. T.; KATRITZKY, A. R. (1996). *Heterocycles in Life and Society*, Wiley, 1997.
100. ROOS-KOZEL, B. L.; JORGENSEN, W. L. (1981) "Computer-Assisted Mechanistic Evaluation of Organic Reactions. 2. Perception of Rings, Aromaticity, and Tautomers". *J. Chem Inf. Comput. Sci.* **21**, 101-111, 1981.
101. ROZENDAAL, E. L. M.; OTT, M. A.; SCHEEREN, H. W. (1994) "A LHASA analysis of taxol". *Recl. Trav. Chim. Pays-Bas*, **113**, 297-303, 1994.
102. RUCKER, G.; RÜCKER, C. (1991) "Computer Perception of Constitutional (Topological) Symmetry. TOPSYM, a Fast Algorithm for Partitioning Atoms and Pairwise Relations among Atoms into Equivalence Classes". *J. Chem. Inf. Comput. Sci.* **31**, 380-386, 1991
103. SALATIN, T.; JORGENSEN, W. (1981) "Computer-Assisted Mechanistic Evaluation of Organic Reactions. Overview." *J. Org. Chem.*, **45**, 2043-2050, 1981.
104. SANDEL, B. B.; SOLOMON, R. W. (1981) "Organic Synthesis Design and the Computer" *J. Chem. Educ.* **58**[10], 198, 1981.
105. SATOH, K.; FUNATSU, K. (1999) "A Novel Approach to Retrosynthetic Analysis Using Knowledge Bases Derived from Reaction Databases". *J. Chem. Inf. Comput. Sci.*, **39**, 316-325, 1999.
106. SEIRAWAN, Y. (1995) *Winning Chess Tactics*, Microsoft Press, 1995.
107. SCHUMMER, J. (1997) "Challenging Standard Distinctions between Science and Technology: The Case of Preparative Chemistry". *HYLE-An International Journal for the Philosophy of Chemistry*, **3**, 81-94, 1997.
108. SCHUMMER, J. (1999) "Coping with the Growth of Chemical Knowledge. Challenges for Chemistry Documentation, Education, and Working Chemists". *Educ. Quim.* **10**[2], 92-101, 1999

109. SEDGEWICK (1990) *Algorithms in C*. Addison-Wesley, 1990.
110. SERRATOSA, F. (1975) *Heuriskô. Introducción a la síntesis orgánica*. Barcelona: Alhambra, 1975.
111. SMIT, W. *et al.* (1998) *Organic Synthesis. The Science behind the Art*. Cambridge: The Royal Society of Chemistry, 1998.
112. SMITH, S.G. (1971) "Computer-Aided Teaching of Organic Synthesis". *J. Chem. Educ.*, **48**, 727-729, 1971.
113. SMITH, M. B. (1990) "Disconnect-by-the-Numbers: A Beginner's Guide to Synthesis". *J. Chem. Educ.* **67**, 848, 1990.
114. SMITH, M. B. (1996) "Disconnect-by-the-Numbers: Protocol for the Synthesis of Alkanes". *J. Chem. Educ.* **73**[4], 304, 1996.
115. STOLOW, R. D.; JONCAS, L. J. (1980) "Computer-Assisted Teaching of Organic Synthesis". *J. Chem. Educ.*, **57**, 868-873, 1980.
116. TAKABATAKE, T.; BERSOHN, M. (1990) "Synthetic routes proposed by a noninteractive synthesis program". *Pure & Appl. Chem.* **62**, 1977-1979, 1990.
117. TAKAHASHI, M. *et al.* (1990) "The Performance of a Noninteractive Synthesis Program". *J. Chem. Inf. Comput. Sci.* **30**, 436-441, 1990.
118. TAKEUCHI, K.; KURODA, C.; ISHIDA, M. (1990) "Prolog-Based Functional Group Perception and Calculation of 1-Octanol/Water Partition Coefficients using Rekker's Fragment Method". *J. Chem. Inf. Comput. Sci.* **30**, 22-26, 1990.
119. The Electronic Nobel Museum Project: The 1990 Nobel Prize in Chemistry.
<http://www.nobel.se/laureates/chemistry-1990-press.html> (consultado en mar/00).
120. TSENG, S. (1997) "Computer-Assisted Reaction Searching Directed toward the Synthesis of Target Molecules". *J. Chem. Inf. Comput. Sci.*, **37**, 1138-1145, 1997.
121. TURRO, N. J. "Geometric and Topological Thinking in Organic Chemistry". *Angew. Chem Int. Ed. Engl.*, **25**, 882-901 (1986).
122. UGI, I.; BAUER; BRANDT; FRIEDRICH; GASTEIGER, J.; JOCHUM; SCHUBERT (1979) "New Applications of Computers in Chemistry". *Angew. Chem Int. Ed. Engl.*, **18**, 111-123, 1979.
123. UGI, I.; BAUER, J.; BIEY, K.; DENGLER, A.; DIETZ, A.; FONTAIN, E.; GRUBER, B.; HERGES, R.; KNAUER, M.; REHSAM, K.; STEIN, N. (1993) "Computer-Assisted Solution of Chemical Problems—The Historical Development and the Present State of the Art of a New Discipline in Chemistry". *Angew. Chem. Int. Ed. Engl.* **32**, 201-227, 1993.
124. VARKONY; SMITH; DJERASSI (1978) "Computer-Assisted Structure Manipulation. Studies in the Biosynthesis of Natural Products" *Tetrahedron* **34**, 841-452, 1978.
125. VOLLMER, J. J. (1983) "Wiswesser Line Notation: An Introduction". *J. Chem. Educ.* **60**[3], 192, 1983.
126. WADE, L. G. (1991) *Química Orgánica*. México: Prentice-Hall, 1993 (segunda edición).
127. WARREN, S. (1978) *Diseño de síntesis orgánica: introducción programada al método del sintón*. Alhambra, 1983.
128. WEININGER (1988) "SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules". *J. Chem. Inf. Comput. Sci.* **28**, 31-36, 1988.
129. WEININGER; WEININGER; WEININGER (1989) "SMILES. 2. Algorithm for Generation of Unique SMILES Notation". *J. Chem. Inf. Comput. Sci.* **29**, 97-101, 1989.
130. WEININGER, D. (1990) "SMILES. 3. Depict. Graphical Depiction of Chemical Structures". *J. Chem. Inf. Comput. Sci.* **30**, 237-243, 1990.
131. WEISE (1990) "Synthesis Simulation by Synthone Substitution". *J. Chem. Inf. Comput. Sci.* **30**, 490-491, 1990.
132. WHITLOCK, H. W. (1998) "On the Structure of Total Synthesis of Natural Products". *J. Org. Chem.* **1998**, **63**, 7982-1989.
133. WINTER, J. H. (1985) "Planning Synthetic Pathways on the Basis of Synthesis Strategies". *J. Chem. Inf. Comput. Sci.* **25**, 389-391, 1985.

134. WIPKE, W. T.; DYOTT, T. M. (1974a) "Simulation and Evaluation of Chemical Synthesis. Computer Representation and Manipulation of Stereochemistry". *JACS*, **96**, 4825-4833, 1974.
135. WIPKE, W. T.; DYOTT, T. M. (1974b) "Stereochemically Unique Naming Algorithm". *JACS* **96**:15, 4834, 1974.
136. WIPKE, W. T. ET AL. (1977) "SECS-Simulation and Evaluation of Chemical Synthesis: Strategy and Planning". en WIPKE Y HOWE (eds), *Computer Assisted Organic Synthesis. ACS Symposium Series* **61**, 1977, p. 97.
137. WIPKE, W. T.; KRISHNAN, S.; OUCHI, G. I. (1978) "Hash Functions for Rapid Storage and Retrieval of Chemical Structures". *J. Chem. Inf. Comput. Sci.* **18**, 32-37, 1978.
138. WIPKE, W. T.; ROGERS, D. (1984a) "Artificial Intelligence in Organic Synthesis SST: Starting Material Selection Strategies and Application of Superstructure Search". *J. Chem. Inf. Comput. Sci.*, **24**, 71-81, 1984.
139. WIPKE, W. T.; ROGERS, D. (1984b) "Rapid Subgraph Search Using Parallelism". *J. Chem. Inf. Comput. Sci.* **24**, 255-262, 1984.
140. YEH, C. (1995) "Isomer Enumeration of Alkanes, Labeled Alkanes, and Monosubstituted Alkanes". *J. Chem. Inf. Comput. Sci.* **35**, 912-913, 1995.
141. ZEFIROV, N. S.; BASKIN, I. I.; PALLYULIN, V. A. (1994) "SYMBEQ Program and its Application in Computer-Assisted Reaction Design". *J. Chem. Inf. Comput. Sci.*, **34**, 994-999, 1994.
142. ZHU, Q.; STILLMAN, M. J. (1996) "Expert Systems and Analytical Chemistry: Recent Progress in the ACExpert Project". *J. Chem. Inf. Comput. Sci.* **36**, 497-509, 1996.
143. ZIMMER, M. H.; CHOPLIN, F.; BONNET, P.; KAUFFMAN, G. (1979) "Automatic Strategy in Computer Design of Synthesis: An Example in Organophosphorous Chemistry". *J. Chem. Inf. Comput. Sci.* **19**, 235-241, 1979.

ÍNDICE

- acetilación, 80
- ácido todomáutico, 80
- ácidos nucleicos, 13
- acoplamiento, reacciones de, 67
- activación, 48
- addatom, 70
- adición, 16, 35
 - de átomos, 70
 - de grupos funcionales *Véase* FGA
 - de Michael. *Véase* Michael, adición
- AIX, 35
- ajedrez, 3, 50
- alcanfor, 10
- Aldrich, 30
- alílico, 63
- Alpha, 27
- alquilación de enolatos, 18, 73
- alquílico, 63
- amonio, cianato de, 8
- análisis
 - conformacional, 11
 - matemático, 13
 - mecanístico, 34
 - retrosintético. *Véase* retrosíntesis
 - sintético, 33
- Angewandte Chemie*, 2
- anillos, 44, 57
 - aromáticos, 61
 - clasificación, 60
 - de interés sintético, 58
 - fusionados, 61
- anisol, 73, 75
- ANSI C, 52
- Apple II+, 85
- applet, 36, 51
- aptitud. *Véase* fitness
- árbol, 19, 64
 - podado. *Véase* podado
- arco, 39
- arista, 39
- aromaticidad, 44, 58
- arte de inventar, 4
- ar-turmerona, 85
- átomo
 - tipos, 63
 - Ayyar-Rao, 80
 - Barone, R., 4, 6, 38, 85
 - bases de datos de reacciones, 34
 - BASIC, 46
 - benceno, 60
 - bencílico, 64
 - Bersohn, M., 31
 - Berthelot, 1
 - Birch, reducción de, 74, 76, 77, 80
 - breakbond, 70
 - brevetoxina B, 12
 - browser, 51
 - buspirona, 21
 - C, 27, 46, 52
 - calificación. *Véase* rating
 - CAMEO, 34, 48, 86
 - canonización, 25, 44, 61
 - CAOS/CAMM Center, 26
 - CASP, 36
 - catalizador, 3
 - catálogos (de materias primas), 37, 45
 - catenanos, 2, 13
 - centro de reacción, 33
 - Chanon, M., 6
 - ChemAxon, 54
 - Chemistry Translator. *Véase* CHMTRN
 - ChemSketch, 38
 - ChemWin, 38
 - CHMTRN, 28
 - ciclización, 25
 - de polienos, 28
 - cicloadición, 25
 - cicloheptanona, 10
 - ciclohexeno, 60, 61
 - ciclopentadienilo, 60
 - ciclopropanación, 28
 - ciclopropano, 60
 - ciencia pura, 2
 - clase, 61
 - clasificación de átomos, 63
 - Clemmensen, reducción de, 76
 - cliente, 51, 53
 - cliente-servidor, 36, 51

combinaciones (en ajedrez), 3
 combinaciones tácticas, 3, 27
 comparación de precursores, 45
 complejidad, 22, 50, 62, 65
 cálculo, 65
 topológica, 22, 44
 Compuestos carbonílicos, 87
 compuestos similares, 37
 computadora, 3
 Computer-Assisted Mechanistic Evaluation of
 Organic Reactions. *Véase* CAMEO
 condensación aldólica, 14, 18, 75
 condiciones de reacción, 48
 condiciones de reacción, 30
 coníferas, 80
 conjunto esencial de anillos esenciales. *Véase*
 ESER
 control cinético/termodinámico, 48
 copia de trabajo, 69
 Corey, E. J., 24, 26
 cortisona, 11
 Couper, 8
 creación de enlaces, 70
 cromatografía, 11
 cubano, 3, 13, 58
 cumulenos, 13

 Desarrollo y Optimización de Procesos de
 Síntesis, 89
 descarboxilación, 75
 desconexión, 37, 66
 tipo -OH, 25
 tipo W, 25
 desenmascaramiento, 28
 deshidratación, 75
 desprotección, 81
 diagrama estructural, 39
 dibujo de estructuras, 38, 53
 Diels-Alder, 16, 18, 28, 44, 78
 distancia química, 33
 distribuida, ejecución, 31
 dodecaedro, 13
 Dugundji-Ugi, 17, 41
 duplicados, 71

 ejecución, 46, 66
 electrones pi, 59
 eliminación, 16, 35
 cíclica, 25
 empuje de electrones, 34
 encabezado, 68
 enfoque de casos, 12
 enlace
 estratégico, 22, 29, 37, 49
 preservado, 22

 enol, 43
 epoxidación, 44
 equinolina, 11
 EROS, 37
 ESER, 58
 espacio fase, 3
 espectroscopía, 11
 esquelético, 66
 Essential Set of Essential Rings. *Véase* ESER
 estado de transición, 11
 imaginario, 17
 estereoquímica, 91
 estereoselectividad, 11
 esterificación, 18, 80
 estrategias, 20, 49
 basadas en grupos funcionales, 28
 basadas en materias primas, 21
 basadas en materias primas, 29, 50
 basadas en transformadas, 21, 28, 50
 de grupos funcionales, 23
 de largo alcance, 28
 estereoquímicas, 23, 29
 oportunistas, 23
 topológicas, 22, 29
 estructura
 acíclica, 22
 cíclica, 22
 políciclica, 22
 representación, 39
 etilenglicol, 74
 evaluación, 29, 37, 56, 71
 Evaluation of Reactions for Organic Synthesis.
 Véase EROS
 exponencial, 1, 5, 20, 43

 Facultad de Química, 87, 89
 farmacia, 2
 fenílicos, 2, 13
 fenílico, 64
 FGA, 19, 25, 67
 FGI, 19, 23, 25, 48, 67, 87
 fitness, 56
 flecha, 15
 fórmula semidesarrollada, 41
 FORTRAN, 27, 35, 46
 FORWARD, 33
 Friedel-Crafts, reacción de acilación, 75
 fuerza bruta, 4
 Fujita, 17, 58
 funcgrp.chm, 64
 función invariante, 61
 furano, 60
 fusiones, 58

 GP0, 67

grafo, 39, 43
 coloreado, 40
 no dirigido, 40
 Grignard, adición de, 78, 79, 83
 grupo
 auxiliar, 23
 protector, 28, 48
 grupo funcional, 15, 43
 enmascarado, 48
 identificación, 64

 halogenación
 alfa a carbono, 18
 por radicales libres, 18
 halolactonización, 28
 Hanser, 57
 Harvard, 24
 helicenos, 13
 Hendrickson, J. B., 16, 17, 31, 47
 heteroátomo, 58
 heterociclo, 44
 aromático, 60
 heurística, 4
 hidratación, 77
 hidrato de carbono, 43
 hidrogenación, 76, 80, 82
 hidrogenación, 74
 hidrólisis, 74, 75, 76, 80
 hidroquinona, 77
 Hippe, Z., 3
 Hoffmann, R., 3
 hoja, 64
 HOLOWIN, 86
 HP, 27
 Hückel, regla de, 58

 Ihlenfeldt W., 36
 índigo, 8
 indol, 60
 industria, 2
 inicialización, 69
 intercambio de grupos funcionales *Véase* FGI
 interfase, 38
 gráfica, 24, 38, 84
 interferencia, 81
 Internet Explorer, 51
 interpretación, 70
 Introducción a la Química Orgánica, 88
 ISIS/Draw, 38
 isohéptica, 19

 jaque mate, 3
Journal of Chemical Education, 84
 juvabiona, 80

 Kekulé, 8

 ladrillos, 21
 Leeds, Reino Unido, 26
 lenguaje de programación, 46
 Ley de Moore, 6
 LHASA, 18, 26, 48, 49, 50, 84
 Linux, 51
 Logic and Heuristics Applied to Synthetic
 Analysis. *Véase* LHASA. *Véase* LHASA
 Logic of Chemical Synthesis, The, 28

 MacOS, 38, 51, 86
 makebond, 70
 Mannich, reacción de, 10, 18
 March, J., 16
 marcianol, 73
 MarvinSketch, 53, 54
 materia prima, 37
 base de datos, 62
 matriz
 de enlace-electrón, 17, 41, 47
 de reacción, 47
 operaciones, 41
 mecanismo, 16, 34
 de reacción, 11
 simbólico, 25
 memoria, 4
 método retrosintético *Véase* retrosíntesis
 Michael, adición de, 18, 75
 microcomputadora, 6, 86
 MOB, 19
 modificación simbólica de grupos funcionales, 25
 modo auxiliar, 67
 modo de análisis, 54, 66
 molécula objetivo *Véase* MOB
 moléculas "exóticas", 2
 MOLFILE, 40
 Moore, G. E., 5
 Morgan, 25, 45
 Mori-Matsui, 82
 MULTISTEP, 85

 naftaleno, 59
 navegación del árbol retrosintético, 54
 Netscape, 51
 nivel de oxidación, 19, 32
 no interactivo, 30
 nodo, 39
 Noordik, J. H., 3
 notación
 lineal, 41, 42
 única, 41
 Notación Lineal de Wiswesser. *Véase* WLN
 numéricos, problemas, 3

número
atómico, 61
de oxidación, 17
primo, 62

objetivo, 15
OCSS, 24
oligomerización, 25
ONESTEP, 85
operaciones
avanzadas, 48
básicas, 38
ORAC, 36
Orf, H. W., 84
Organic Chemistry Simulation of Synthesis.
Véase OCSS
Organic Synthesis Exploration Tool. *Véase* OSET
ORGSYN, 84
OSET, 51, 53, 80, 87
pruebas de campo, 89
Ott, M. A., 3
oxidación, 16

Pappus, 13
parámetro, 66
paso, 15
PDP, 24, 26
penicilina V, 11
peptido, 21
percepción, 42, 56
de precursores, 70
pericíclicas, reacciones, 16
piridina, 60
pirrol, 60
plotter, 24
podado, 49
búsqueda de ramas con formas específicas, 49
de ramas que no van en cierta dirección, 50
difuso, 50
por límite de profundidad, 49
temprano, 49

polihedranos, 2
posiciones (en ajedrez), 3
posprocesamiento, 70
precursor, 15
problema educativo, 6
producto, 15
programas de CAOS, 24
de seguridad generación, 36
procedimientos
de simulación de reacciones
de simulación de reacciones
de simulación de reacciones
de simulación de reacciones

púrpura de Tiro, 8

quinina, 11
quinolina, 60
quinona, 59

R, 43
radialenos, 13, 59
radicales libres, 16
rama, 64
rapidez de cálculo, 4
rating, 29, 69, 71
razonamiento mecanístico, 34
reacción, 15
genérica, 37
inversa, 14
REACCS, 34, 36
reactivo, 15
reactivos selectivos, 11
reducción, 16
Reformatsky, reacción de, 62
remoción
de átomos, 70
de grupos funcionales, 18
rendimiento, 29, 49
representación canónica, 45
reserpina, 11
resonancia, 59
retrón, 15, 21
entorno del retrón, 46
identificación, 68
parcial, 16
tipo, 18
retrosíntesis, 12
REVIEW, 85
revisiones sobre CAOS, 24
rangs .chm, 60
Robinson, R., 10
reacción de anillación, 18, 28, 44, 79
rotaxanos, 13
ruptura de enlaces, 70
rxas.chm, 66

sacrificio (en ajedrez), 3, 50
Sandel, B. B., 84
Schummer, 1
SECs, 36
secundario, 67
selectividad, 11
de simulación de reacciones
de simulación de reacciones
de simulación de reacciones
de simulación de reacciones

Silicon Graphics, 27
simbólicos, problemas, 3
simetría estructural (topológica), 25, 45
simplificación, 65, 71
Simplified Molecular Input Line Entry System.
Véase SMILES

síntesis
combinatoria, 12
con soporte sólido, 12
convergente, 32, 49
ideal, 13, 16, 31
orgánica, 8
paralelizada, 49
sistema operativo, 51
sm.chm, 62
SMILES, 42
canónico, 45
y compuestos aromáticos, 60

Smith, S. G., 84
software libre, 7
SOS, 85
Sparc20, 31
Stolow, R. D., 85
subestructura, 15
subgrafo, 43
Sun Microsystems, 31
sustitución, 16, 35
electrofilica aromática, 58, 74
nucleofilica aromática, 58
SYNCHEM, 30
SynGen, 31, 49, 86
SYNLIB, 34, 36
Synthesis Generator. *Véase* SynGen
SynTree, 86

tabla
de átomos, 25, 40
de enlaces, 25, 40, 44
tableta Rand, 24
tácticas, 48
tautómero, 35
taxol, 12
TCP/IP, 55
tecnología, 2
teoría de grafos, 19
teoría estructural, 8
todomatsu, 81
Transfer Control Protocol/Internet Protocol. *Véase*
TCP/IP
Trotter, D. J., 86
Trotter, D. J., 86

aplicación, 69
auxiliar, 16, 18
base de datos, 46, 51
clasificación formal, 17
de anillos, 18, 69
de dos grupos funcionales, 18, 69
de patrón arbitrario, 18
de un grupo funcional, 18, 69
objetivo, 21
representación, 46
representación abstracta, 47
representación algorítmica, 46
simplificativa, 16, 18
subobjetivo, 21
tipos, 16, 68
tropilio, 60
tropinona, 10
Turner, 13

Ugi, I., 47
UNAM, 87
Universidad Nacional Autónoma de México.
Véase UNAM
UNIX, 27, 35
urea, 8

variables automáticas, 69
VAX, 26, 35
vértice, 39
vitamina A, 11
VMS, 35

Weininger, 61
Wholer, F., 8
Williamson, síntesis de éteres, 77
Willstater, 10
Windows, 38, 51, 86
Wittig, reacción de, 18, 77, 83
WLN, 30, 41
WODCA, 36
Wolff-Kishner, reducción de, 76
Workbench for the Organization of Data for
Chemical Applications. *Véase* WODCA

X, 38
x-terminol, 10

Esta tesis fue realizada entre septiembre de 1999 y marzo de 2001. Para su realización se requirieron aproximadamente 36 kg de chocolate en polvo y 550 L de leche entera, además de una cantidad indeterminada de galletas.