ELEMENTS OF C

Computer Science

Tokens & Syntax

- **The compiler collects the characters of a program into tokens.**
  - **Tokens make up the basic vocabulary of a computer language.**
- **The compiler then checks the tokens to see if they can be formed into legal strings according to the syntax (the grammar rules) of the language.**

Characters Used in C Programs

- Lowercase letters
  - a b c . . . z
- Uppercase letters
  - A B C . . . Z
- Digits
  - 0 1 2 3 4 5 6 7 8 9
- Other characters
  - + - * / = ( ) { } [ ] < > ` "
  - ! @ # $ % & _ ^ ~ \ . , ; : ?
- White space characters
  - blank, newline, tab, etc.

The Six Kinds of Tokens in ANSI C

- **Keywords**
- **Identifiers**
- **Constants**
- **String Constants**
- **Operators**
- **Punctuators**

Keywords

- **Keywords are C tokens that have a strict meaning.**
  - **They are explicitly reserved and cannot be redefined.**
- **ANSII C has 32 key words.**
  - **Some implementations such as Borland's C or Microsoft's C have additional key words.**

ANSII C Keywords

`auto`	`do`	`goto`	`signed`	`unsigned`
`break`	`double`	`if`	`sizeof`	`void`
`case`	`else`	`int`	`static`	`volatile`
`char`	`enum`	`long`	`struct`	`while`
`const`	`extern`	`register`	`switch`	
`continue`	`float`	`return`	`typedef`	
`default`	`for`	`short`	`union`	

Identifiers

- **An identifier is a token:**
  - **Composed of a sequence of letters, digits, and the underscore character _**
    - **Note: Variable names are identifiers**
- **Lower- and uppercase letters are treated as distinct.**
- **Identifiers should be chosen so that they contribute to the readability and documentation of the program.**

Special Identifiers

- **main**
  - **C programs always begin execution at the function main.**
- **Identifiers that begin with an underscore should be used only by systems programmers**
  - **Because they can conflict with system names.**

The Length of Discriminated Identifiers

- **On older systems only the first eight characters of an identifier are discriminated.**
  - **identifier_one and identifier_two would be the same identifier.**
- **In ANSI C, at least the first 31 characters of an identifier are discriminated.**

Constants

- **Integer Constants**
  - **25 and 0**
- **Floating Constants**
  - **3.14159 and 0.1**
- **Character Constants**
  - **‘a’ and ‘B’ and ‘+’ and ‘;’ but not “a” or “B”**

Special Character Constants

- **The backslash is called the escape character.**
  - **The newline character '\n' represents a single character called newline.**
  - **Think of \n as "escaping" the usual meaning of n.**
- **Enumeration constants will be discussed later in the course.**

String Constants

- **A sequence of characters enclosed in a pair of double quote marks, such as “abc” is a string constant, or a string literal.**
- **Character sequences that would have meaning if outside a string constant are just a sequence of characters when surrounded by double quotes.**
- **String constants are treated by the compiler as tokens and the compiler provides the space in memory to store them.**

Is it a String or Not a String?

- **“this is a string constant”**
- **“” /* the null string */**
- **“   “ /* a string of blanks */**
- **“ a = b + c; “ /* is not executed */**
- **“ /* this is not a comment */ “**
- **/* “ this is not a string “ */**
- **“ and**

  neither is this “
- **‘a’ /* a character, not a string */**

The Mathematical Operators

- **We looked at the mathematical operators briefly in the 3rd class:**

  + - * / %

- **In a C program we typically put white space around binary operators to improve readability.**

  a + b rather than a+b

The sizeof Operator

- **The C sizeof unary operator if used to find the number of bytes needed to store an object.**
  - **sizeof(object) returns an integer that represents the number of bytes needed to store the object in memory.**

printf()

printf(control string, other arguments);

- **The expressions in other_arguments are evaluated and converted according to the formats in the control string and are then placed in the output stream.**

printf("%-14sPayRate: $%-4.2f\n", " Rohan Kumar", 9.95);

Rohan Kumar Pay Rate: $9.95

- **Characters in the control string that are not part of a format are placed directly in the output stream.**

The Formats in the Control String

printf("Get set: %d %s %f %c%c\n",

1, "two", 3.33, 'G', 'O');

- %d Print 1 as a decimal number
- %s Print "two" as a string
  - "string" means a sequence of characters.
- %f Print 3.33 as a float
  - decimal or floating-point number
- %c Print 'G' & '0' as characters.

printf() Conversion Characters

Conversion character	How the corresponding argument is printed
c	as a character
d,i	as a decimal integer
u	as an unsigned decimal integer
o	as an unsigned octal integer
x,X	as an unsigned hexadecimal integer
e	as a floating-point number: 7.123000e+00
E	as a floating-point number: 7.123000E+00
g	in the shorter of the e-format or f-format
G	in the shorter of the E-format or f-format
s	as a string
p	the corresponding argument is a pointer to void; it prints as a hexadecimal number.
n	argument is a pointer to an integer into which the number of characters written so far is printed; the argument is not converted.
%	with the format %% a single % is written; there is no corresponding argument to be converted.

printf( ) Conversion Specifications

- **field width (optional)**
  - **An optional positive integer**
  - **If the converted argument has fewer characters than the specified width, it will be padded with spaces on the left or right depending on the left or right justification.**
  - **If the converted argument has more characters, the field width will be extended to whatever is required.**
- **precision (optional)**
  - **Specified by a period followed by a nonnegative integer.**
  - **Minimum number of digits to be printed for d, i, o, u, x, and X conversions.**
  - **Minimum number of digits to the right of the decimal point for e, E, and f conversions.**
  - **Maximum number of significant digits for G and g conversions.**
  - **Maximum number of characters to be printed for an s conversion.**

printf ( ) Example

printf("Get set: %d %s %f %c%c\n",
1, "two", 3.33, 'G', 'O');

The first argument is the control string

"Get set: %d %s %f %c%c\n"

The formats in the control string are matched (in order of occurrence) with the other arguments.

Use of printf ( )

- **printf( ) is used for printing output. When printf( ) is called it is passed a list of arguments of the form:**

control string & other arguments

- **The arguments to printf( ) are separated by commas.**

Errors in printf ( ) Formats

- A floating point format in a printf ( ) statement is of the form %m.nf
  - The value of m specifies the field width, not the number of digits to the left of the decimal point.
  - The value of n specifies the number of digits to the right of the decimal point.
- To specify two decimal digits to the left of the decimal point and three to the right, use %6.3f.

Use of scanf()

- **scanf() is analogous to printf(), but is used for input rather than output.**
  - **scanf()in a program stops the execution of the program while you type something in from the keyboard.**

scanf ( ) Arguments

- **The first argument is a control string with formats similar to those used with printf().**
  - **The formats determine how characters in the input stream (what you are typing) will be interpreted so they can be properly stored in memory.**

Scanf ( )'s Other Arguments

- After the control string, the other arguments are addresses.
- Example: assume x is declared as an integer variable.

  scanf("%d", &x);

  The & is the address operator. It says "store the value entered at the address of the memory location named x".

scanf ( ) Conversion

Conversion How characters in the Character input stream are converted.

c	Character
d	decimal integer
f	floating-pint number (float)
lf	floating-point number (double)
Lf	floating-point number (long double)
s	string

A Peculiarity of scanf ( )

- **With printf() the %f format is used to print either a float or a double.**
- **With scanf() the format %f is used to read in a float, and %lf is used to read in a double.**

Another scanf() Peculiarity

- **When reading in numbers, scanf() will skip white space characters (blanks, newlines, and tabs).**
- **When reading characters, white space is not skipped.**

The Return Value of scanf()

- When the scanf() function reads in data typed by a user, it returns the number of successful conversions.
  - scanf("%d%d%d", &first, &second, &third);
    - Should return a value 3 if the user correctly types three integers.
    - Suppose the user enters 2 integers followed by a string -- what happens?
      - What does our system do?

Common Programming Errors

- **Failure to correctly terminate a comment.**
- **Leaving off a closing double quote character at the end of a string.**
- **Misspelling or not declaring a variable.**
- **Misspelling a function name.**
- **Omitting the ampersand (&) with scanf( ).**

How the Compiler Handles Comments

/* This is a comment */

The compiler first replaces each comment with a single blank.

Thereafter, the compiler either disregards white space or uses it to separate tokens.

System Considerations

- **Syntax (Compile -Time) Errors**
  - **Syntax errors are caught by the compiler.**
  - **The compiler attempts to identify the error and display a helpful error message.**
- **Run-Time Errors**
  - **Errors that occur during program execution.**
  - **Memory errors caused by not using the address operator & with a scanf ( ) argument.**

Style

- Use white space and comments to make your code easier to read and understand.
  - Indent logical subgroups of code by 3 spaces.
- Choose variable names that convey their use in the program.
- Place all #includes, #defines, main()s, and braces { } -- that begin and end the body of a function -- in column 1.

Number Systems

Common Number Systems

System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, … 9	Yes	No
Binary	2	0, 1	No	Yes
Octal	8	0, 1, … 7	No	No
Hexa-decimal	16	0, 1, … 9, A, B, … F	No	No

Quantities/Counting (1 of 3)

Decimal	Binary	Octal	Hexa-decimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7

Quantities/Counting (2 of 3)

Decimal	Binary	Octal	Hexa-decimal
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

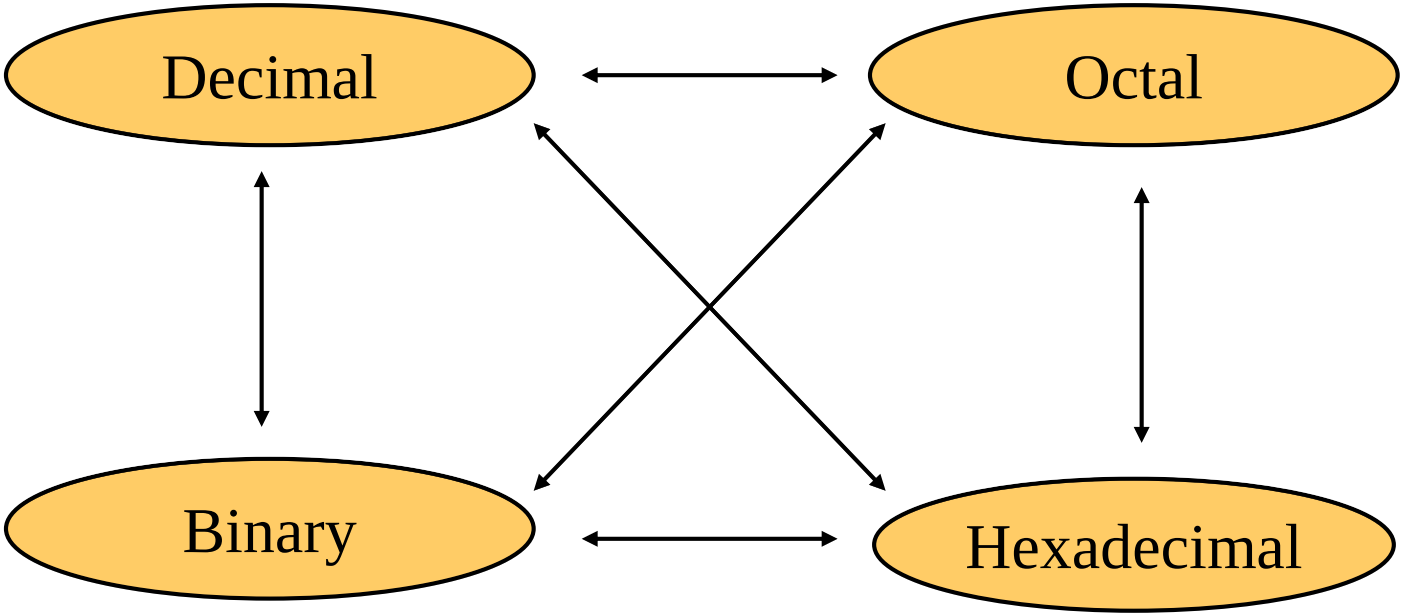
Quantities/Counting (3 of 3)

Decimal	Binary	Octal	Hexa-decimal
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17

Etc.

Conversion Among Bases

- The possibilities:

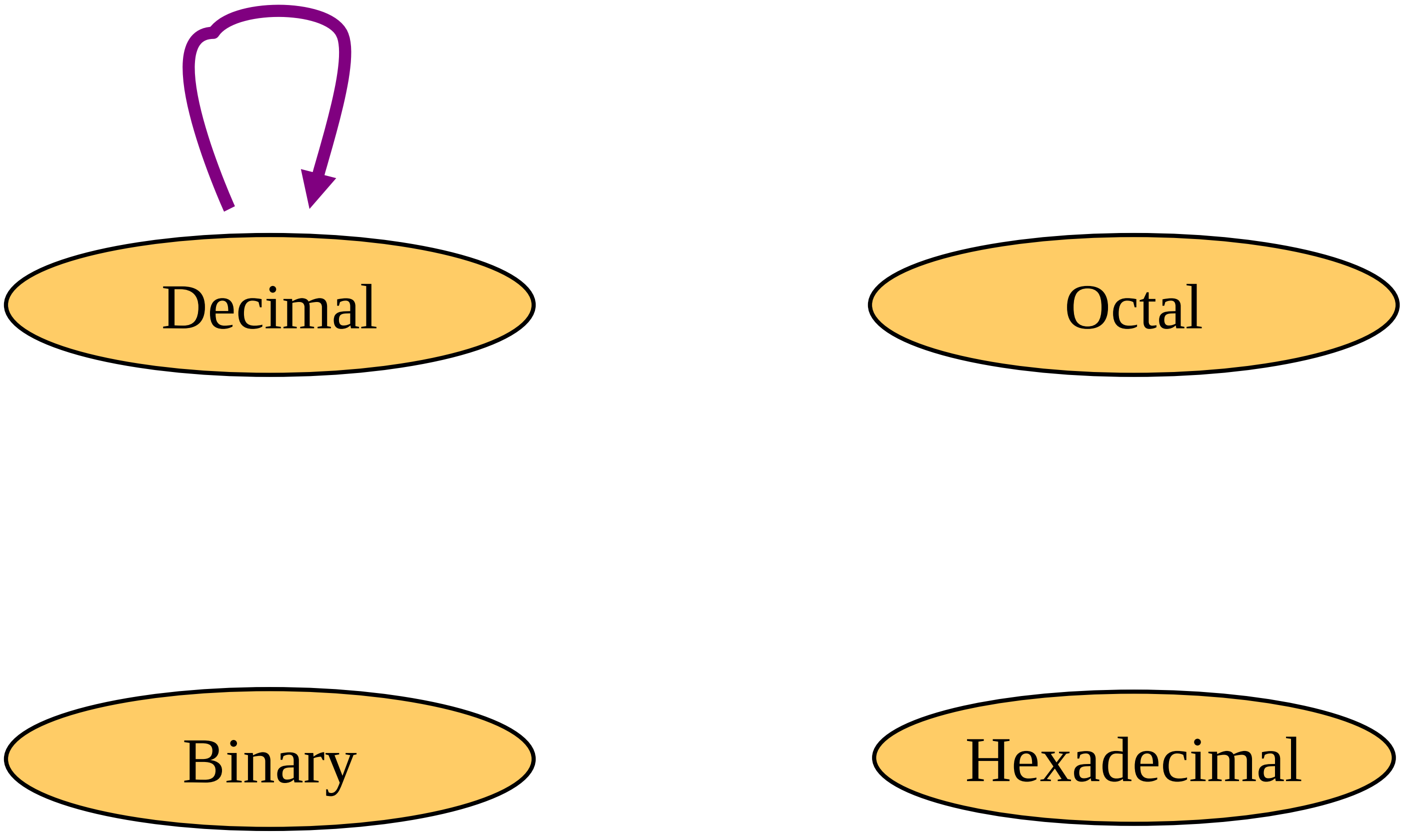


Quick Example

$$25_{10} = 11001_2 = 31_8 = 19_{16}$$

Base

Decimal to Decimal (just for fun)

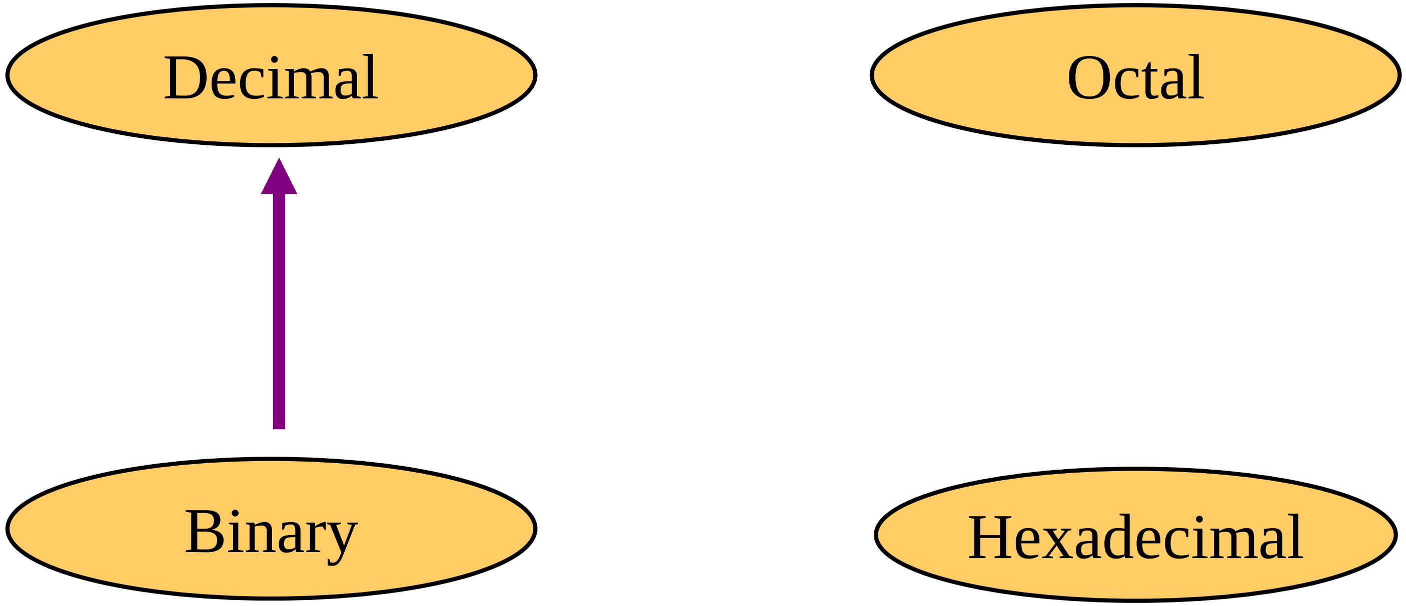


Weight

$$
\begin{aligned}
125_{10} \Rightarrow \quad & 5 \times 10^0 = 5 \\
& 2 \times 10^1 = 20 \\
& 1 \times 10^2 = 100 \\
\hline
& \qquad\qquad 125
\end{aligned}
$$

Base

Binary to Decimal



Binary to Decimal

- Technique
  - Multiply each bit by 2^n, where n is the “weight” of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

Bit “0”

101011_2 =>

$1 \times 2^0 = 1$

$1 \times 2^1 = 2$

$0 \times 2^2 = 0$

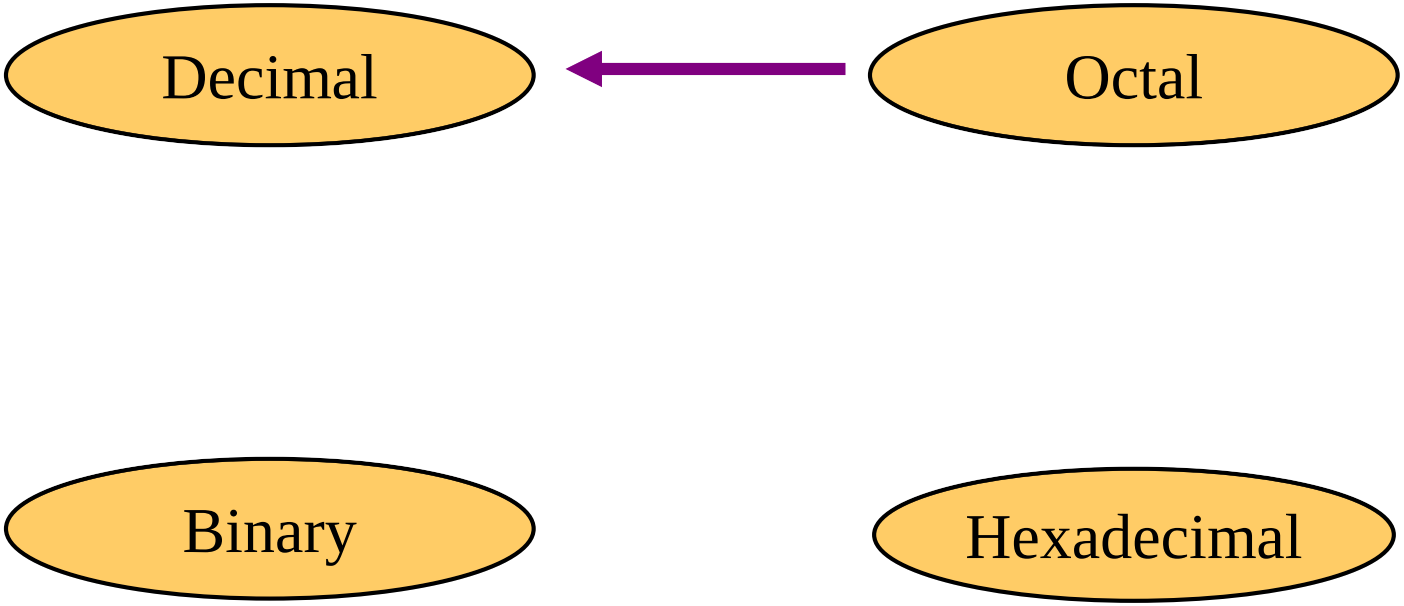
$1 \times 2^3 = 8$

$0 \times 2^4 = 0$

$1 \times 2^5 = 32$

43_{10}

Octal to Decimal



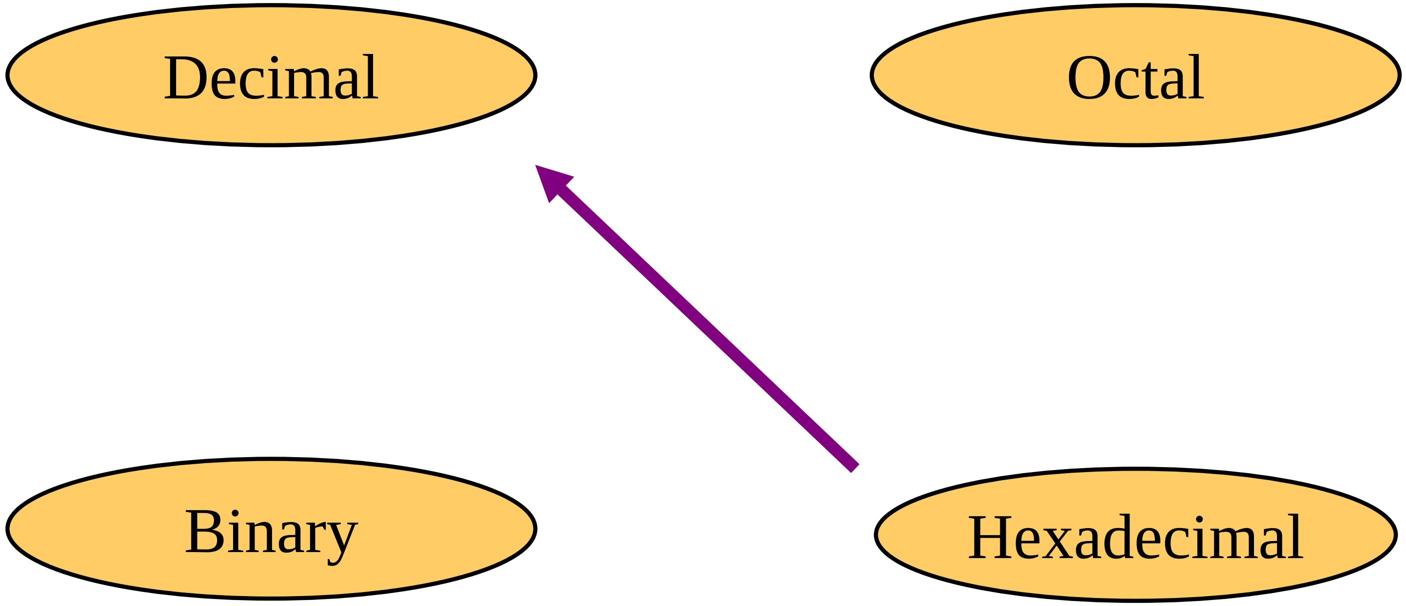
Octal to Decimal

- Technique
  - Multiply each bit by 8^n, where *n* is the “weight” of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

$$
\begin{array}{lllr}
724_8 \Rightarrow & 4 \times 8^0 = & & 4 \\
& 2 \times 8^1 = & & 16 \\
& 7 \times 8^2 = & & 448 \\
\hline
& & & 468_{10}
\end{array}
$$

Hexadecimal to Decimal



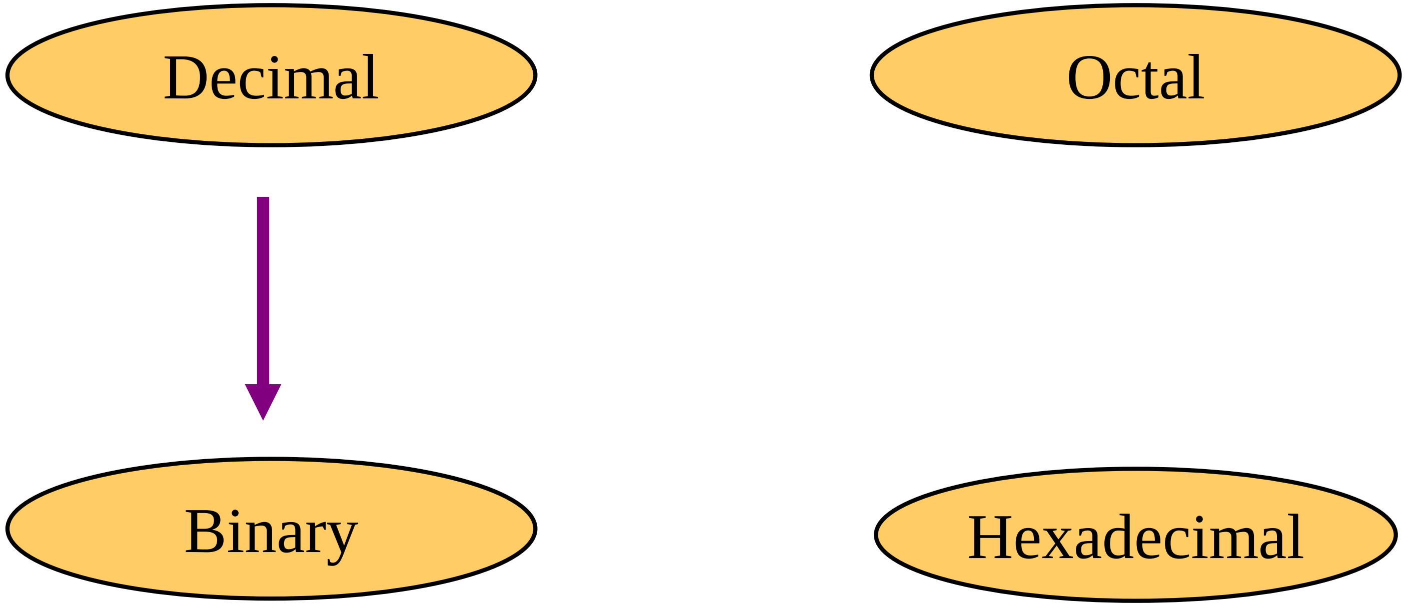
Hexadecimal to Decimal

- Technique
  - Multiply each bit by 16n, where n is the “weight” of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

$$
\begin{array}{lllllll}
ABC_{16} \Rightarrow & C \times 16^0 & = 12 & \times & 1 & = & 12 \\
 & B \times 16^1 & = 11 & \times & 16 & = & 176 \\
 & A \times 16^2 & = 10 & \times & 256 & = & \underline{2560} \\
 & & & & & & 2748_{10}
\end{array}
$$

Decimal to Binary



Decimal to Binary

- Technique
  - Divide by two, keep track of the remainder
  - First remainder is bit 0 (LSB, least-significant bit)
  - Second remainder is bit 1
  - Etc.

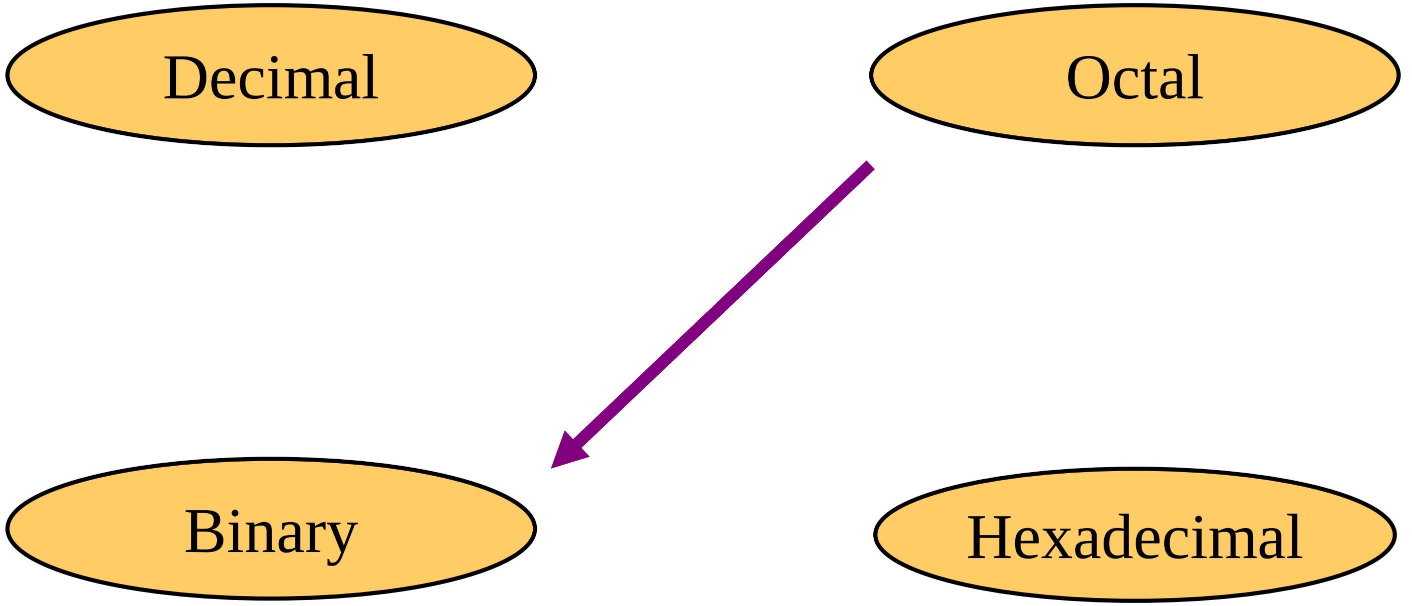
Example

$125_{10} = ?_2$

2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1

$125_{10} = 1111101_2$

Octal to Binary



Octal to Binary

- Technique
  - Convert each octal digit to a 3-bit equivalent binary representation

Example

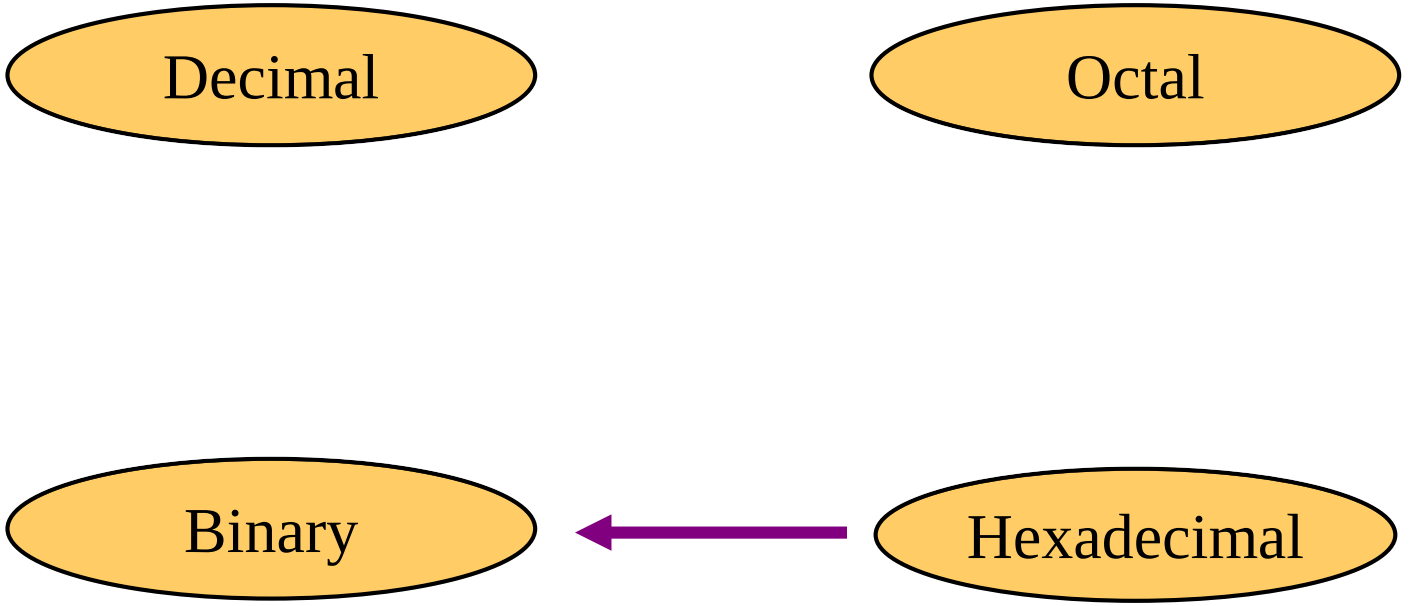
$705_8 = ?_2$

7 0 5

111 000 101

$705_8 = 111000101_2$

Hexadecimal to Binary



Hexadecimal to Binary

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation

Example

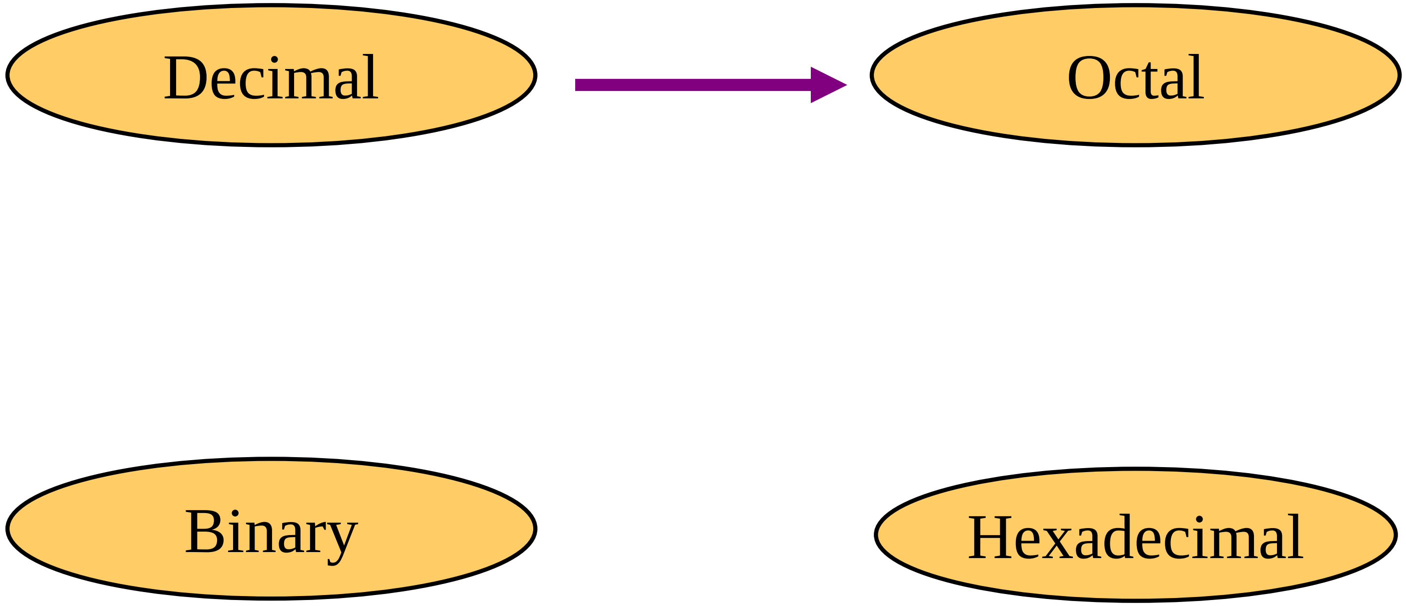
$10AF_{16} = ?_2$

1 0 A F

0001 0000 1010 1111

$10AF_{16} = 0001000010101111_2$

Decimal to Octal



Decimal to Octal

- Technique
  - Divide by 8
  - Keep track of the remainder

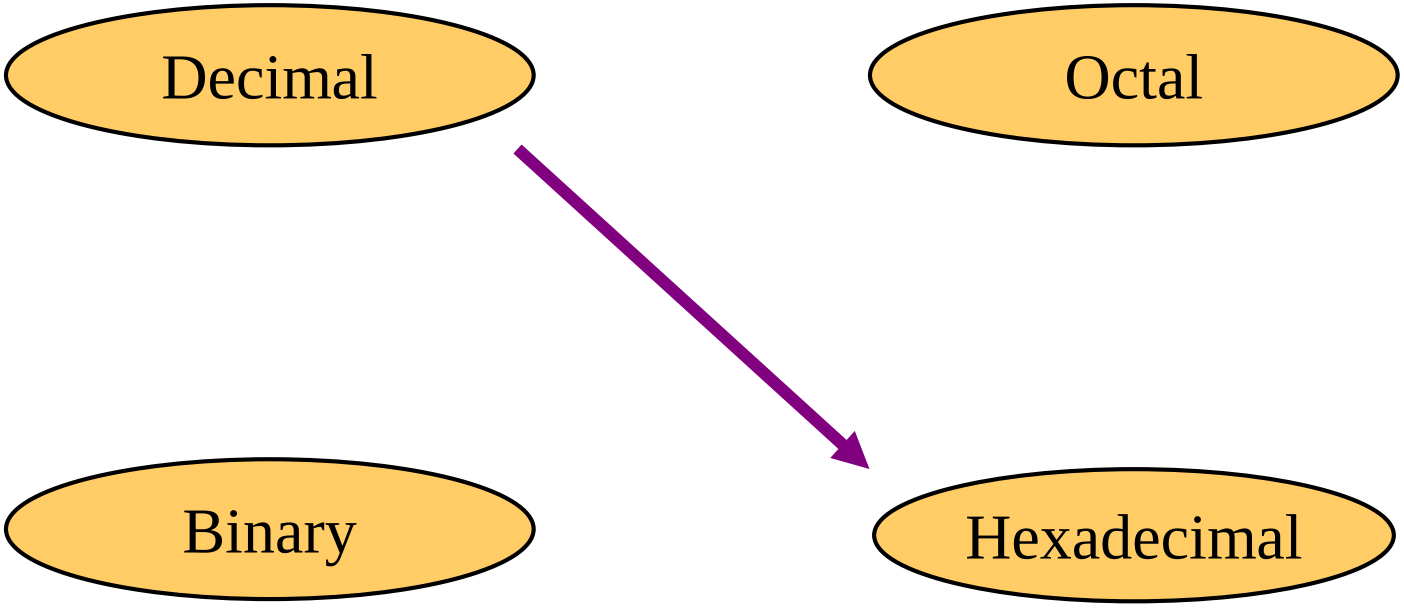
Example

$1234_{10} = ?_8$

8	1234	
8	154	2
8	19	2
8	2	3
	0	2

$1234_{10} = 2322_8$

Decimal to Hexadecimal



Decimal to Hexadecimal

- Technique
  - Divide by 16
  - Keep track of the remainder

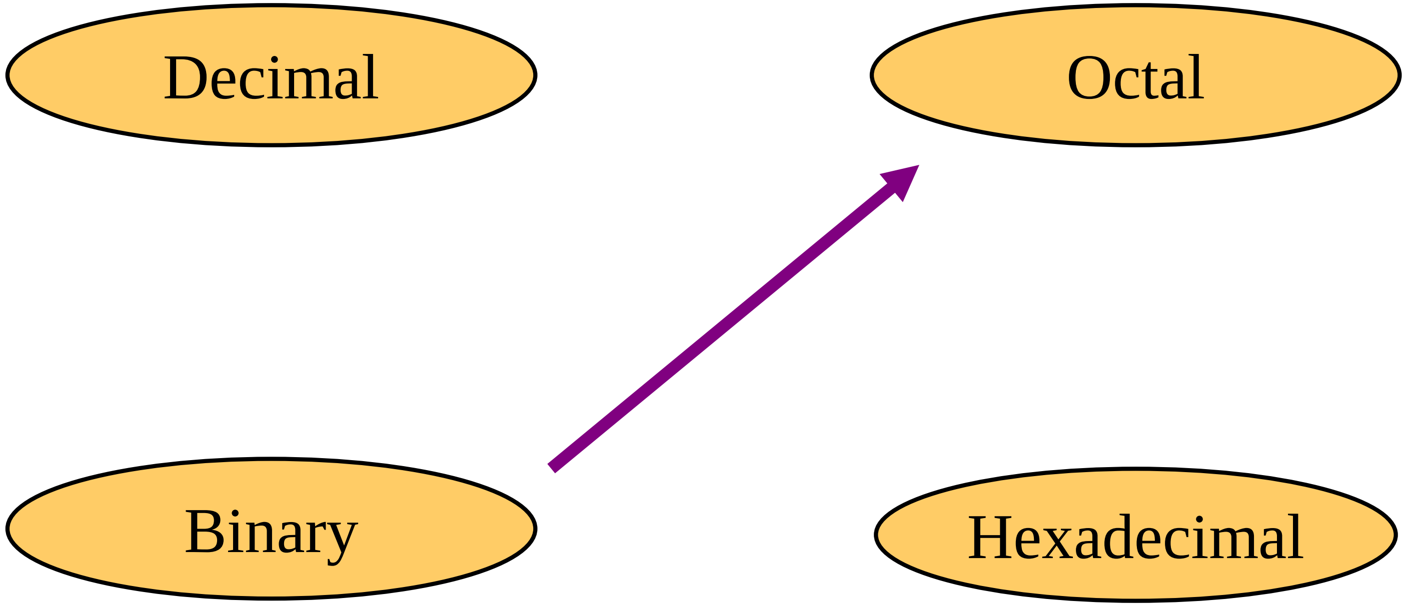
Example

$1234_{10} = ?_{16}$

16 | 1234
77 | 16 2
4 | 16 13 = D
0 4

$1234_{10} = 4D2_{16}$

Binary to Octal



Binary to Octal

- Technique
  - Group bits in threes, starting on right
  - Convert to octal digits

Example

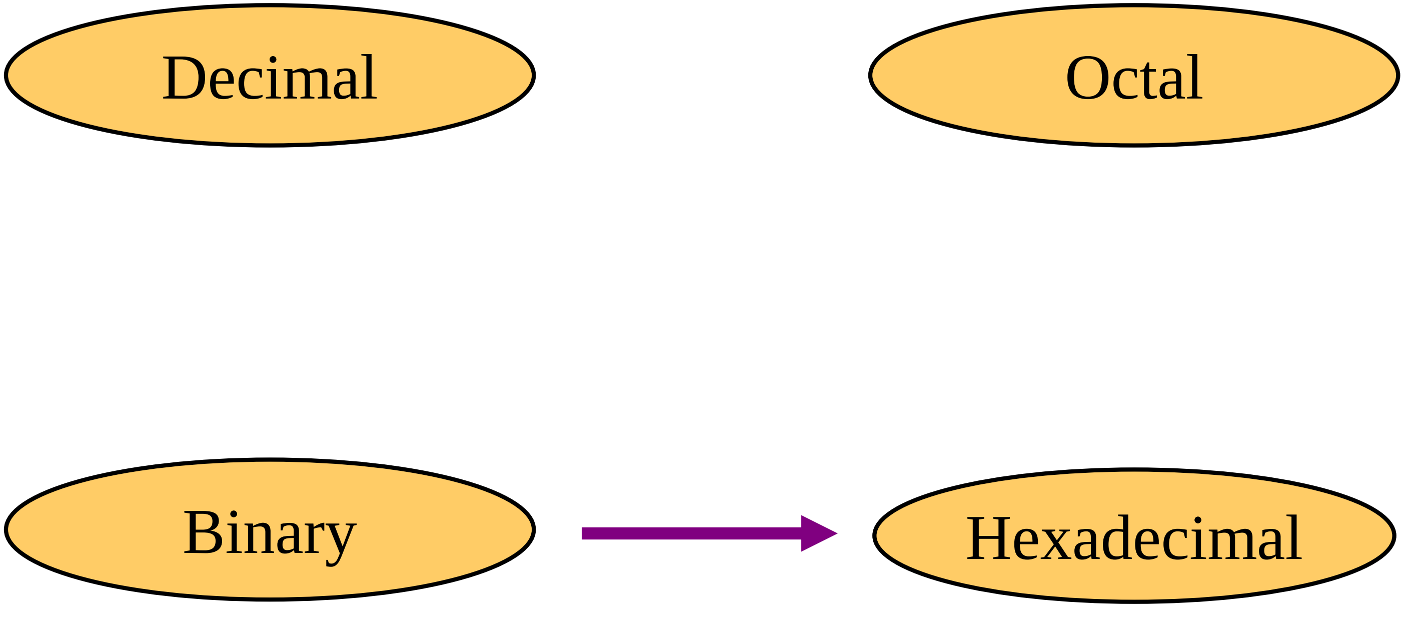
$1011010111_2 = ?_8$

1 011 010 111

1 3 2 7

$1011010111_2 = 1327_8$

Binary to Hexadecimal



Binary to Hexadecimal

- Technique
  - Group bits in fours, starting on right
  - Convert to hexadecimal digits

Example

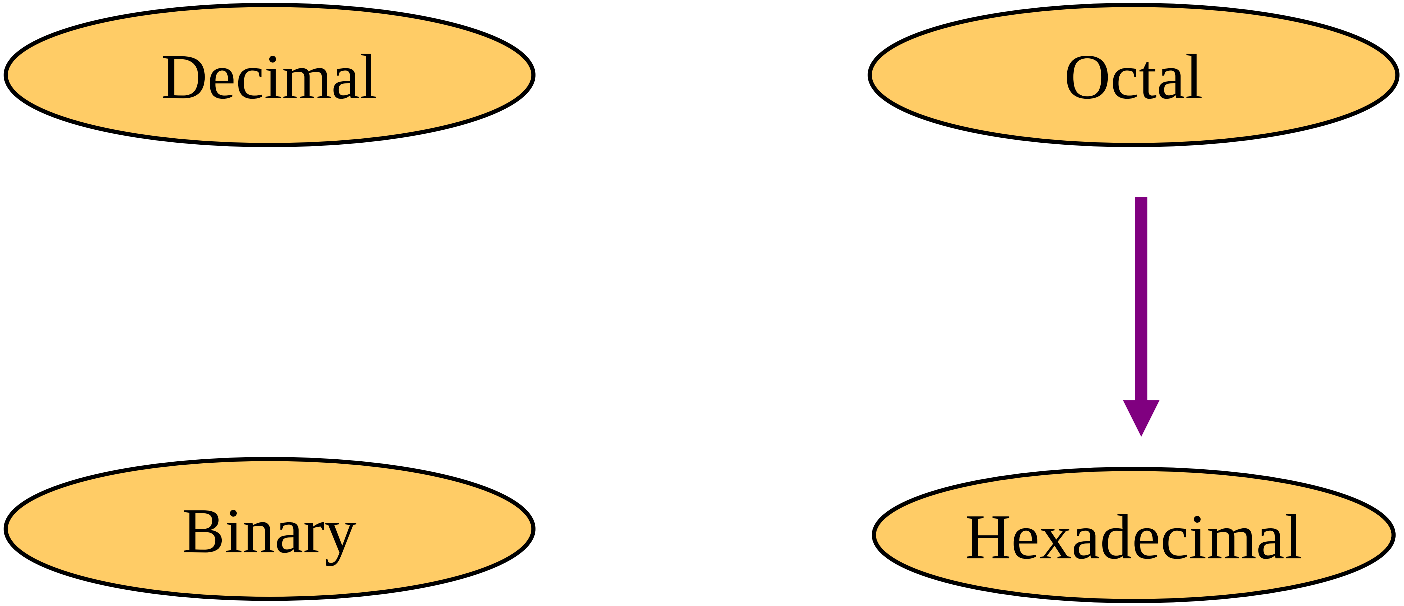
$1010111011_2 = ?_{16}$

10 1011 1011

2 B B

$1010111011_2 = 2BB_{16}$

Octal to Hexadecimal

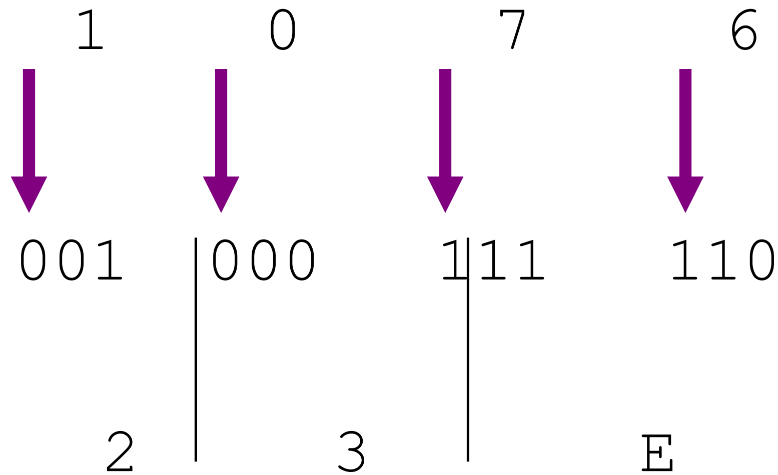


Octal to Hexadecimal

- Technique
  - Use binary as an intermediary

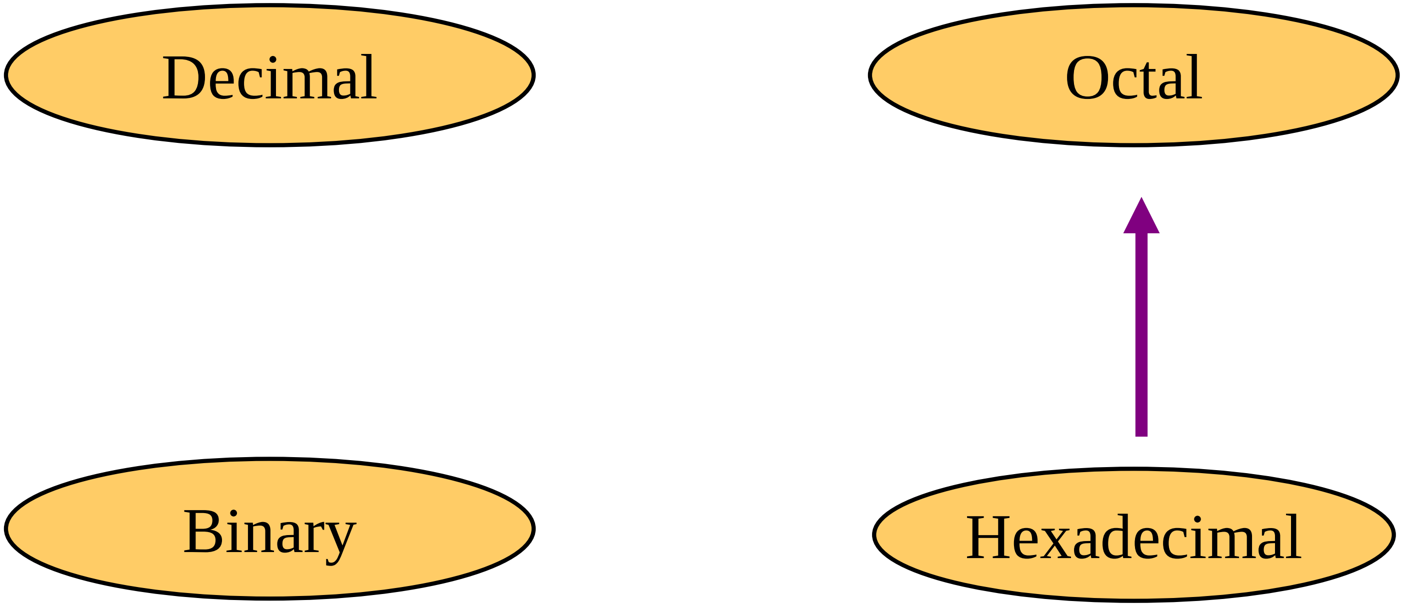
Example

$1076_8 = ?_{16}$



$1076_8 = 23E_{16}$

Hexadecimal to Octal



Hexadecimal to Octal

- Technique
  - Use binary as an intermediary

Example

$1F0C_{16} = ?_8$

1 F 0 C

0001 1111 0000 1100

1 | 7 | 4 | 1 | 4

$1F0C_{16} = 17414_8$

Exercise – Convert ...

Decimal	Binary	Octal	Hexa-decimal
33			
	1110101		
		703	
			1AF

Answer

Exercise – Convert …

Answer

Decimal	Binary	Octal	Hexa-decimal
33	100001	41	21
117	1110101	165	75
451	111000011	703	1C3
431	110101111	657	1AF

Fractions

- Decimal to decimal

$$
3.14 \Rightarrow \quad
\begin{array}{r}
4 \times 10^{-2} = 0.04 \\
1 \times 10^{-1} = 0.1 \\
3 \times 10^{0} \;\; = 3 \\
\hline
3.14
\end{array}
$$

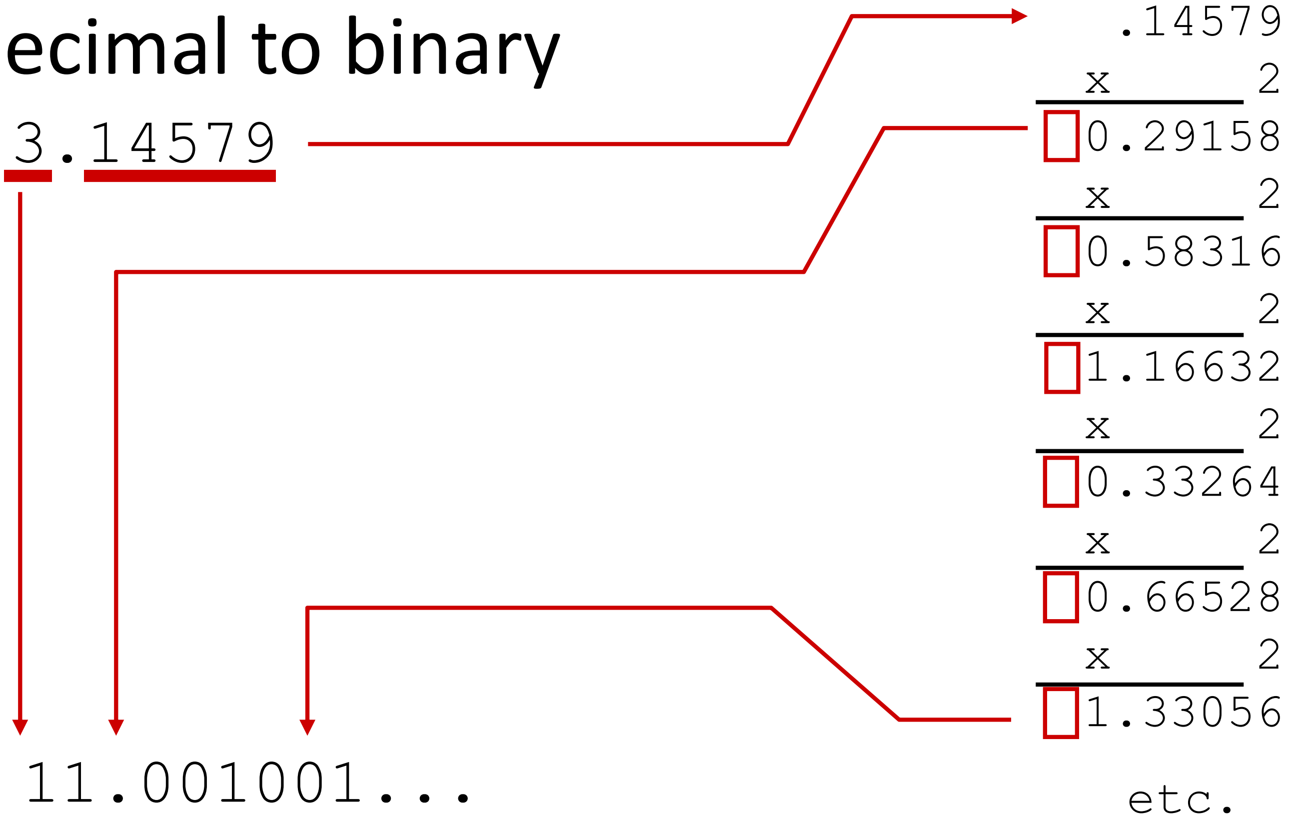
Fractions

- Binary to decimal

$$
\begin{array}{lrcl}
10.1011 \Rightarrow & 1 \times 2^{-4} & = & 0.0625 \\
 & 1 \times 2^{-3} & = & 0.125 \\
 & 0 \times 2^{-2} & = & 0.0 \\
 & 1 \times 2^{-1} & = & 0.5 \\
 & 0 \times 2^{0} & = & 0.0 \\
 & 1 \times 2^{1} & = & \underline{2.0} \\
 & & & 2.6875
\end{array}
$$

Fractions

- Decimal to binary



Exercise – Convert ...

Decimal	Binary	Octal	Hexa-decimal
29.8			
	101.1101		
		3.07	
			C.82

Answer

Exercise – Convert …

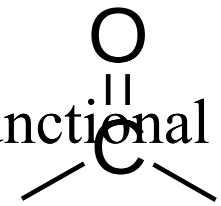
Answer

Decimal	Binary	Octal	Hexa-decimal
29.8	11101.110011…	35.63…	1D.CC…
5.8125	101.1101	5.64	5.D
3.109375	11.000111	3.07	3.1C
12.5078125	1100.10000010	14.404	C.82

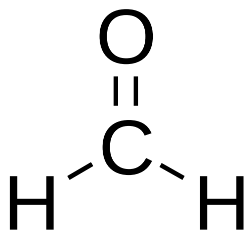
Aldehydes and Ketones

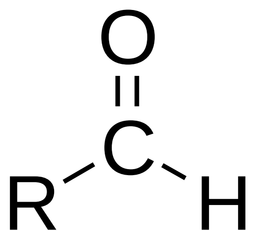
Chemistry

ALDEHYDES AND KETONES

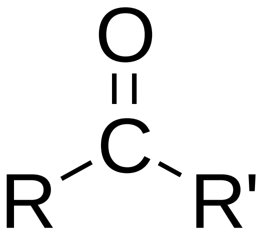
“carbonyl” functional group: 

Aldehydes

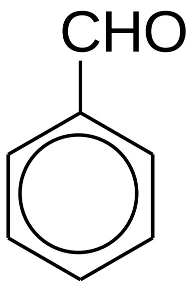




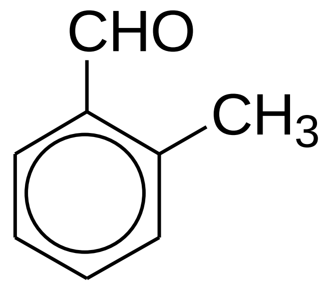
Ketones



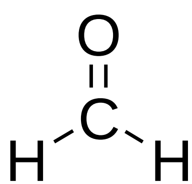
R can be Ar



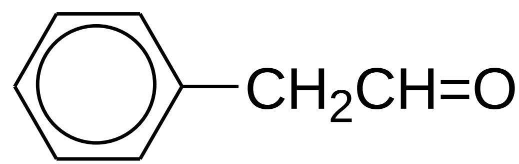
benzaldehyde



o-tolualdehyde



formaldehyde



phenylacetaldehyde

Nomenclature:

Aldehydes, common names:

Derived from the common names of carboxylic acids;

drop –ic acid suffix and add –aldehyde.

$CH_3CH_2CH_2CH{=}O$

butyraldehyde

$(CH_3)_2CHCH{=}O$

isobutyraldehyde
(α-methylpropionaldehyde)

Aldehydes, IUPAC nomenclature:

Parent chain = longest continuous carbon chain containing the carbonyl group; alkane, drop –e, add –al. (note: no locant, -CH=O is carbon #1.)

$CH_3CH_2CH_2CH{=}O$

butanal

$$\begin{array}{c} \quad\; CH_3 \\ \quad\;| \\ CH_3CHCH{=}O \end{array}$$

2-methylpropanal

$H_2C{=}O$

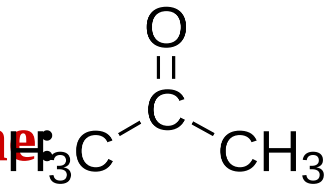
methanal

$CH_3CH{=}O$

ethanal

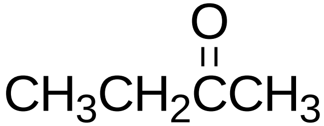
Ketones, common names:

Special name:

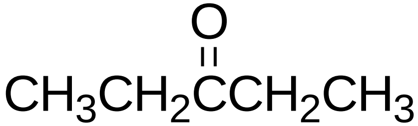


acetone

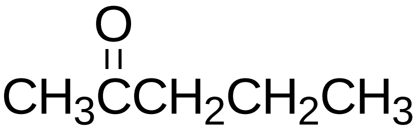
"alkyl alkyl ketone" or "dialkyl ketone"



ethyl methyl ketone

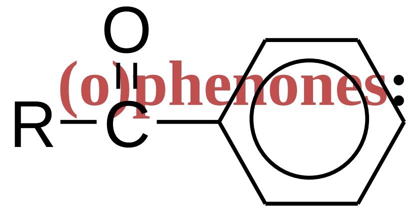


diethyl ketone

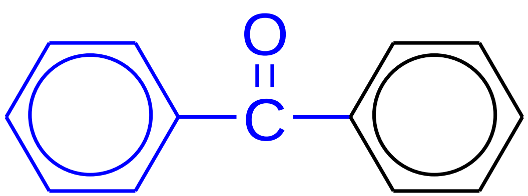


methyl *n*-propyl ketone

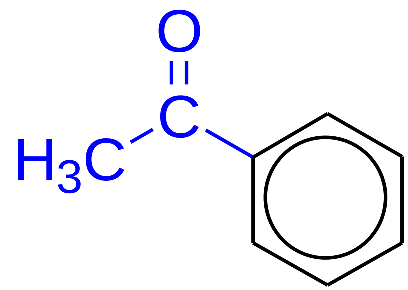
(o)phenones:



Derived from common name of carboxylic acid, drop –ic acid, add –(o)phenone.



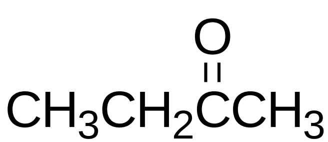
benzophenone



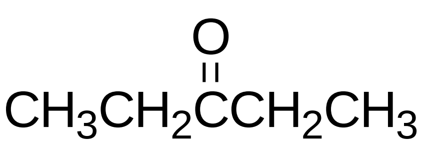
acetophenone

Ketones: IUPAC nomenclature:

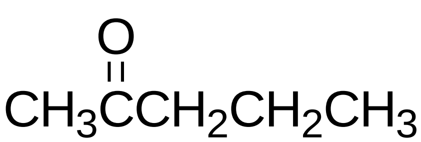
Parent = longest continuous carbon chain containing the carbonyl group. Alkane, drop –e, add –one. Prefix a locant for the position of the carbonyl using the principle of lower number.



2-butanone

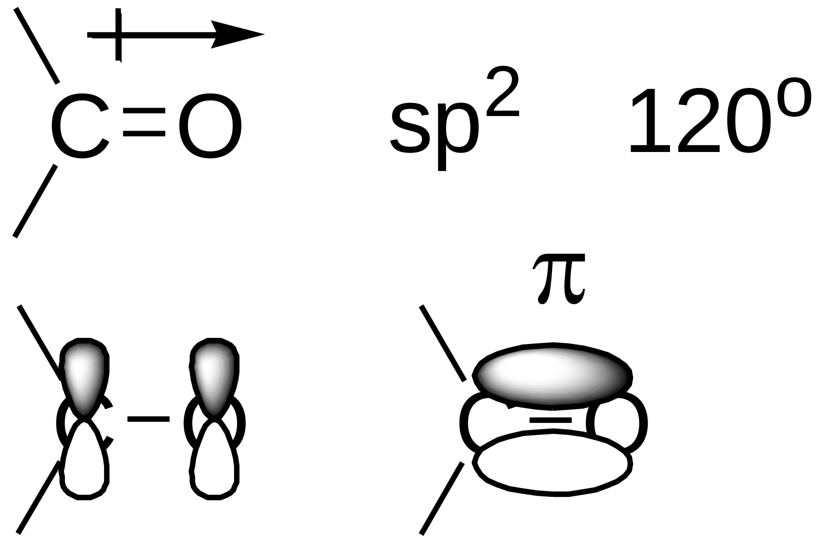


3-pentanone



2-pentanone

Physical properties:



Polar

no hydrogen bonding

Melting point /boiling point are relatively moderate for covalent substances

water insoluble except four-carbons or less)

Aldehydes synthysis:

1. Oxidaton of 1º alcohols
2. Oxidation of methyl benzene
3. Reduction of acid chlorides

Ketones synthysis:

1. Oxidation of 2º alcohols
2. Friedel-Crafts acylation
3. Coupling of R_2CuLi with acid chloride

Aldehydes synthesis

1) oxidation of primary alcohols:

$RCH_2\text{-}OH \quad + \quad K_2Cr_2O_7$, special conditions → RCH=O

$RCH_2\text{-}OH \quad + \quad C_5H_5NHCrO_3Cl$ → RCH=O

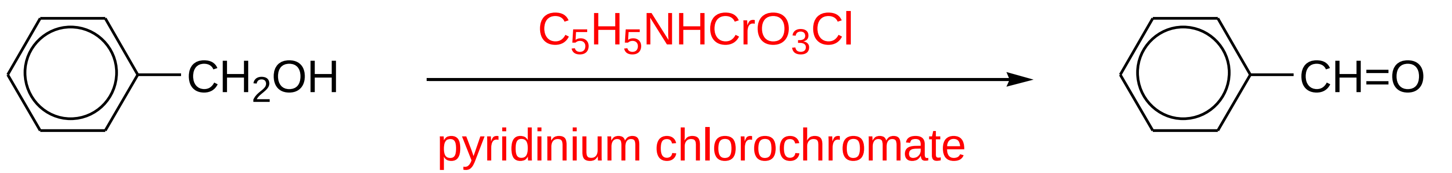
(pyridinium chlorochromate)

[**With other oxidizing agents, primary alcohols → RCOOH**]

$$CH_3CH_2CH_2CH_2CH_2OH \xrightarrow{K_2Cr_2O_7\text{, special conditions!}} CH_3CH_2CH_2CH_2CH{=}O$$

1-pentanol

pentanal
valeraldehyde



benzyl alcohol

benzaldehyde

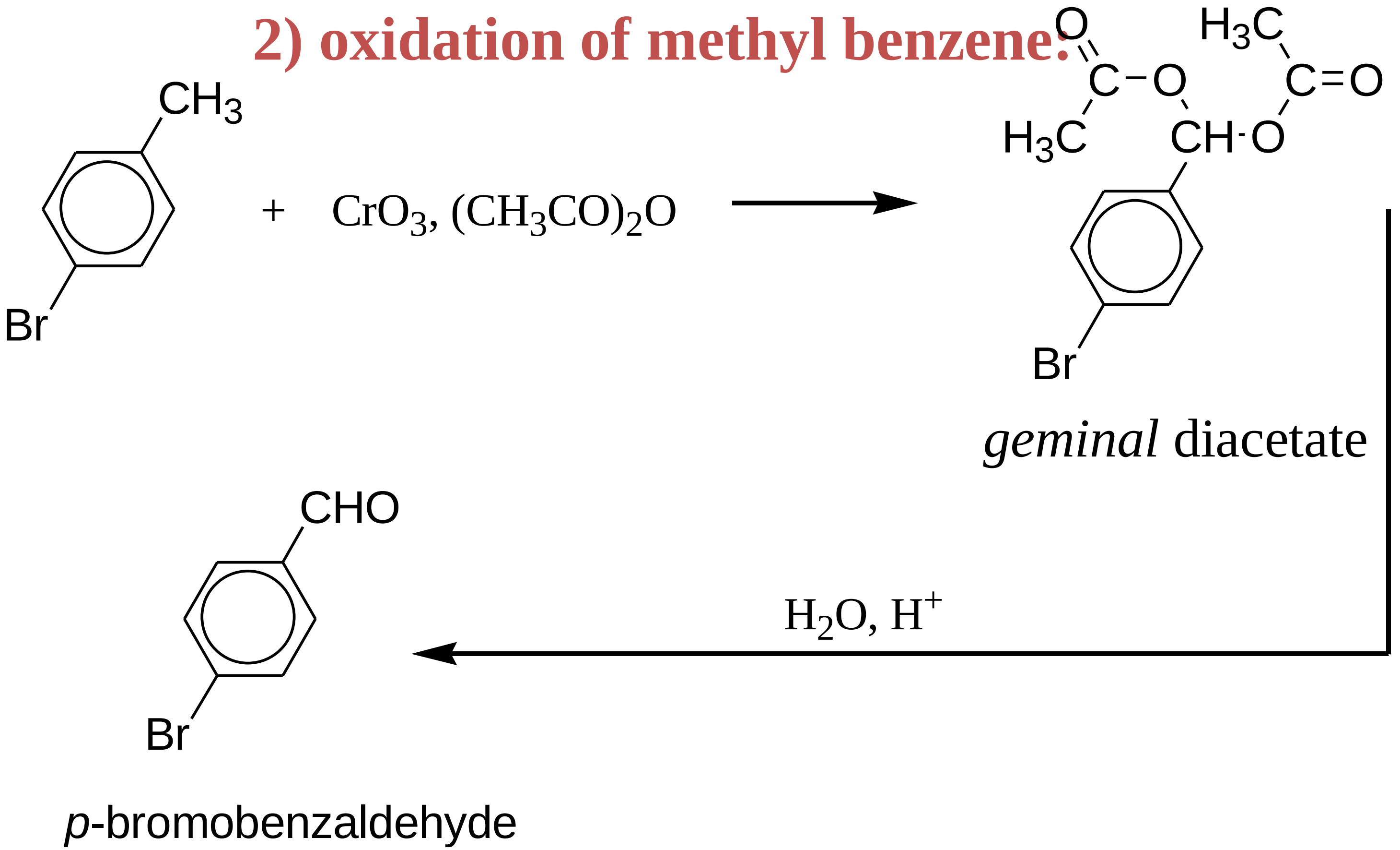
$$\left\{ CH_3CH_2CH_2CH_2CH_2OH + K_2Cr_2O_7 \longrightarrow CH_3CH_2CH_2CH_2CO_2H \right\}$$

1-pentanol

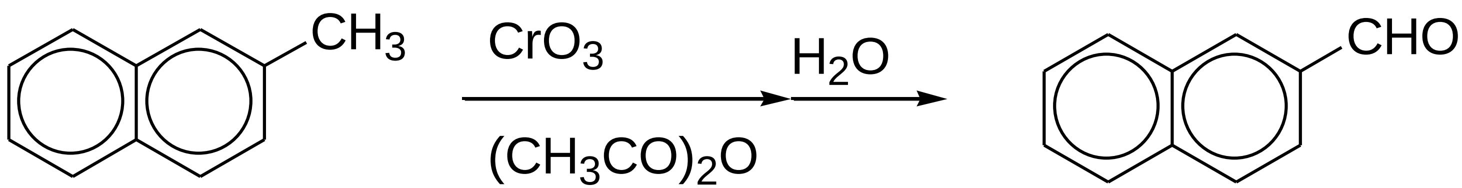
pentanoic acid

Aldehyde synthesis:

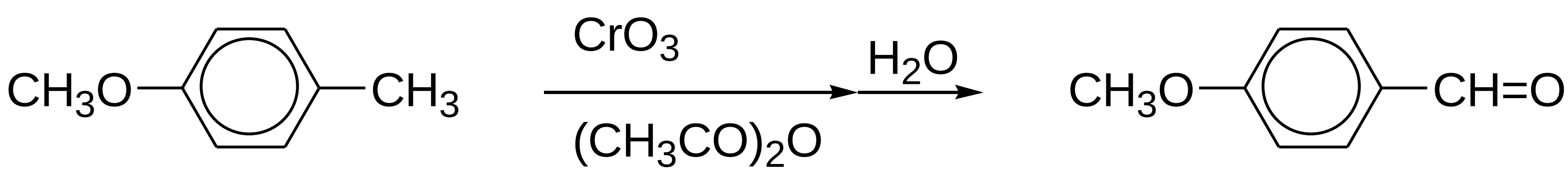
2) oxidation of methyl benzene:



<u>Aromatic</u> aldehydes only!



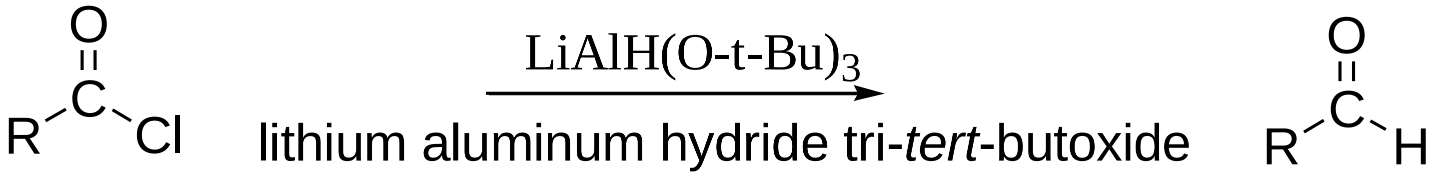
2-methylnaphthalene 2-naphthaldehyde

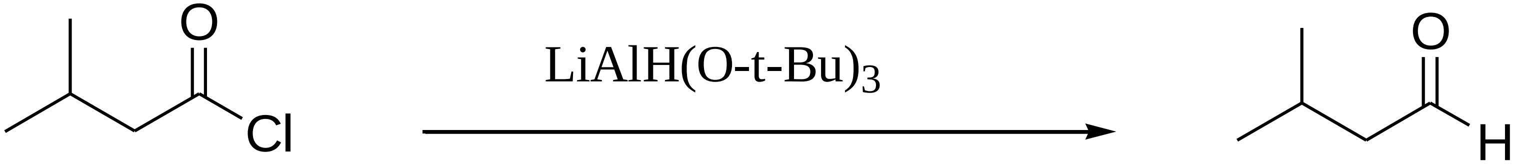


p-methylanisole *p*-anisaldehyde

Aldehyde synthesis:

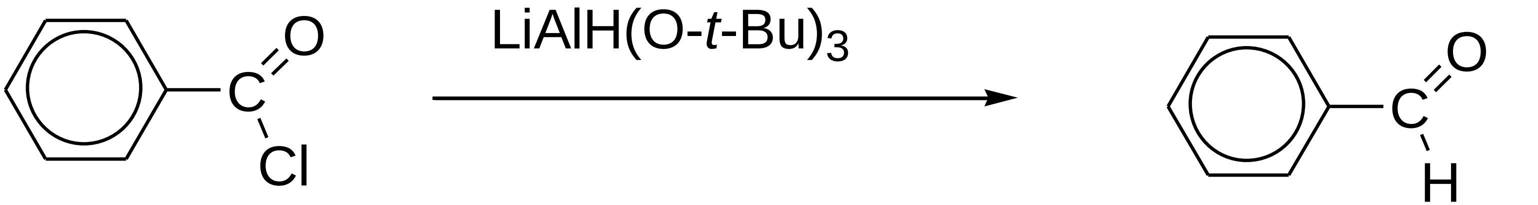
3) reduction of acid chloride





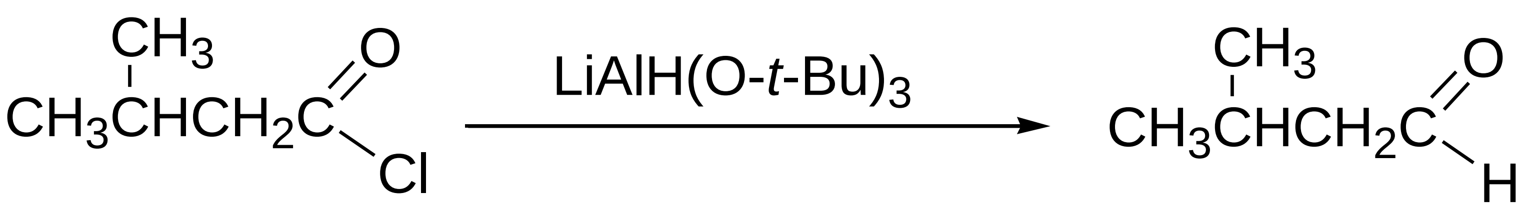
isovaleryl chloride

isovaleraldehyde



benzoyl chloride

benzaldehyde

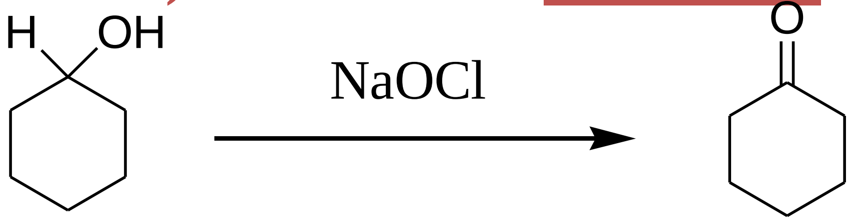


isovaleryl chloride

isovaleraldehyde

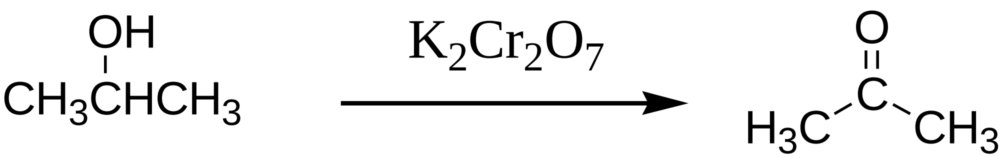
Ketone synthesis:

1) oxidation of secondary alcohols



cyclohexanol

cyclohexanone



isopropyl alcohol

acetone

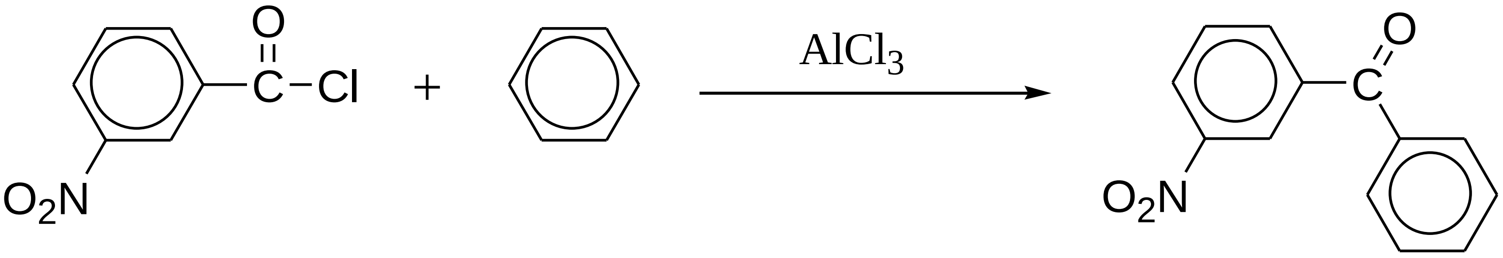
Ketone synthesis:

2) Friedel-Crafts acylation

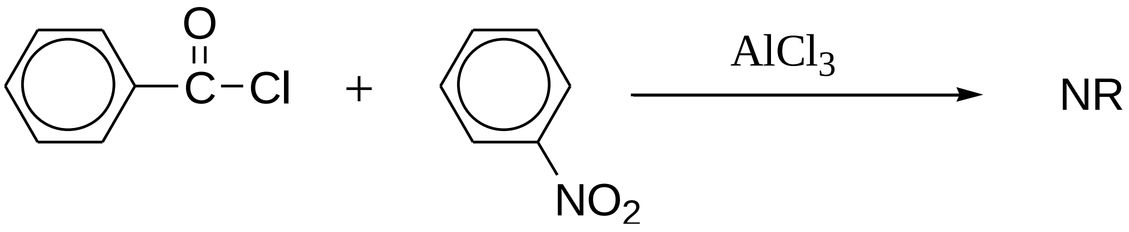
$$RCOCl, AlCl_3 \; + \; ArH \xrightarrow{AlCl_3} R{-}\overset{\overset{\displaystyle O}{\|}}{C}{-}Ar \; + \; HCl$$

butyrophenone

Aromatic ketones (phenones) only!

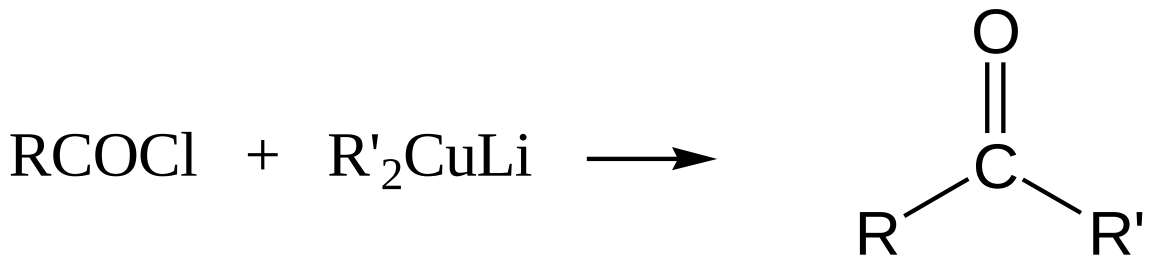


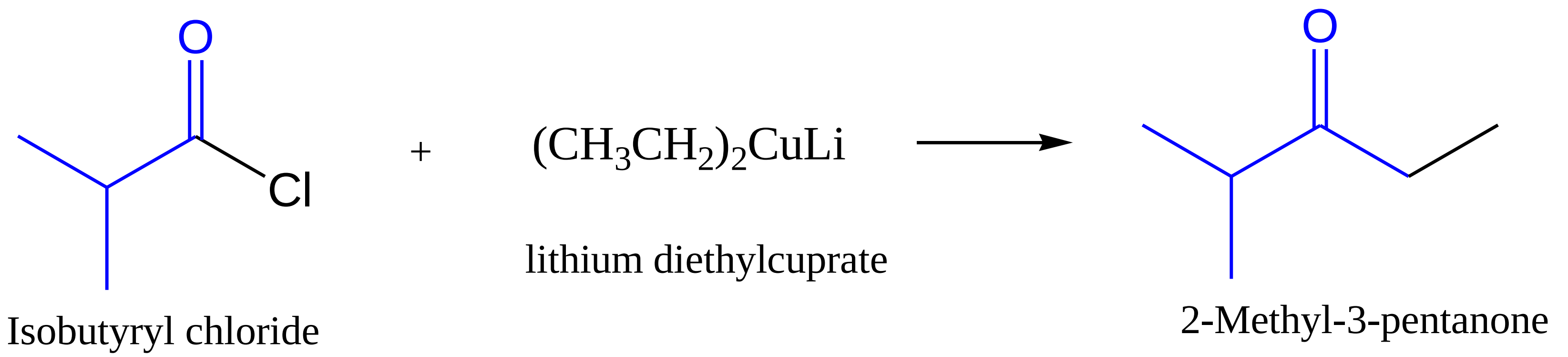
m-nitrobenzophenone

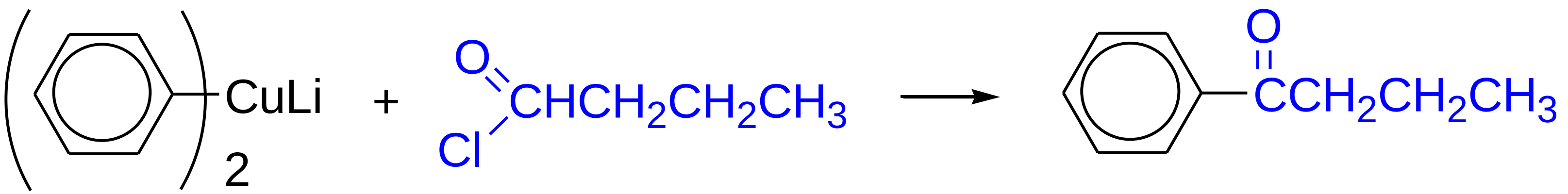


Friedel Crafts acylation does not work on deactivated rings.

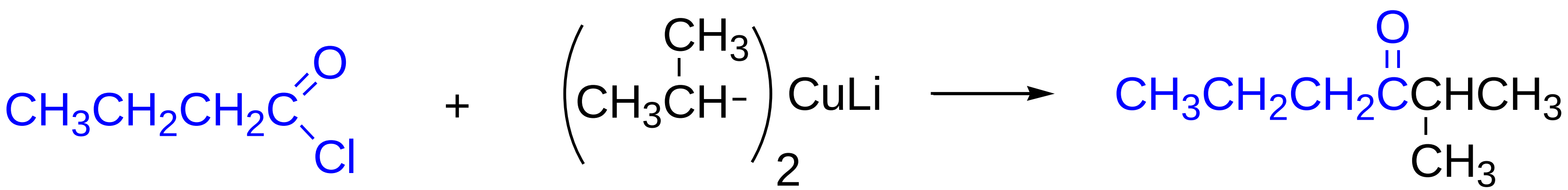
Ketone synthesis: 3) coupling of RCOCl and R_2CuLi





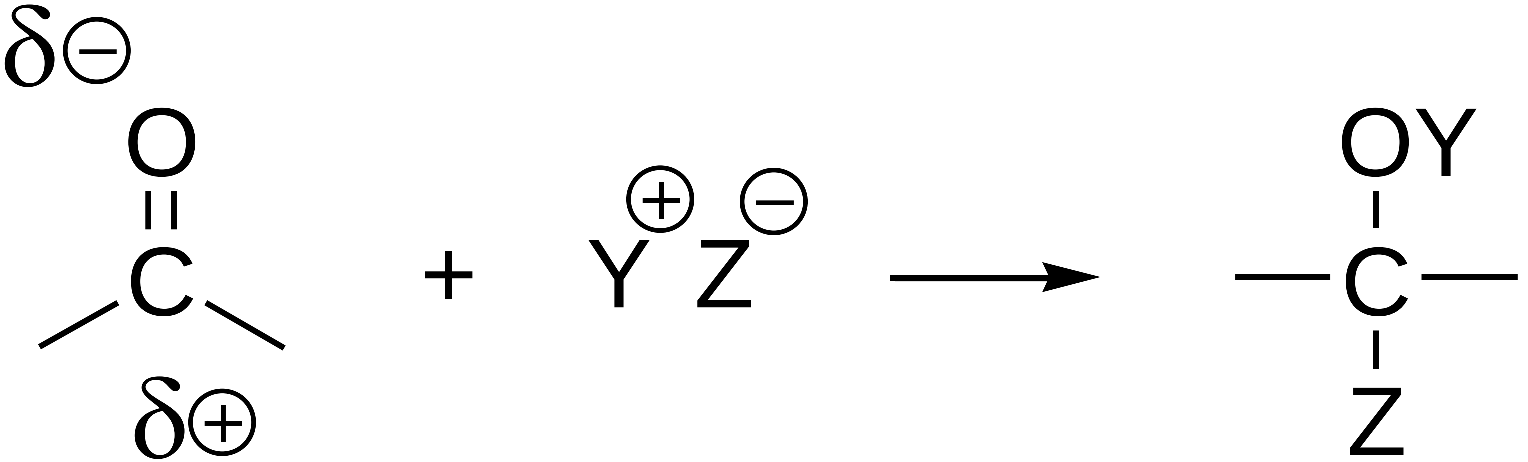


butyrophenone



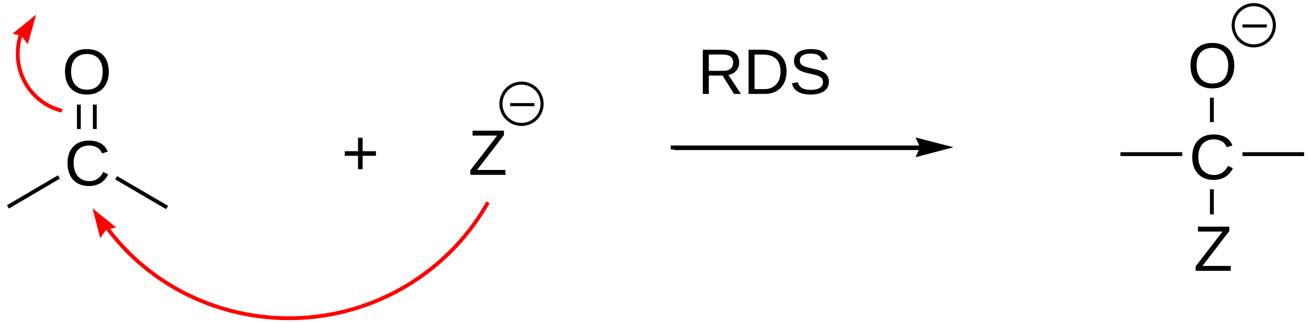
2-methyl-3-hexanone

Nucleophilic addition to carbonyl:

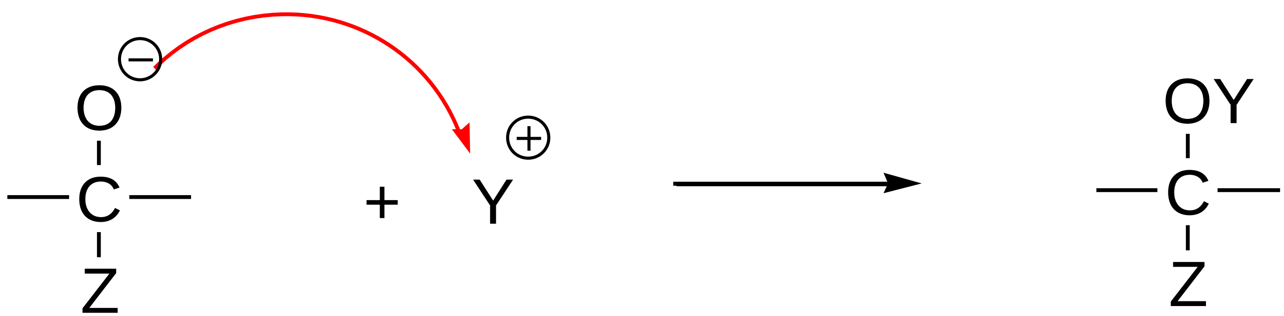


Mechanism: nucleophilic addition to carbonyl

1)

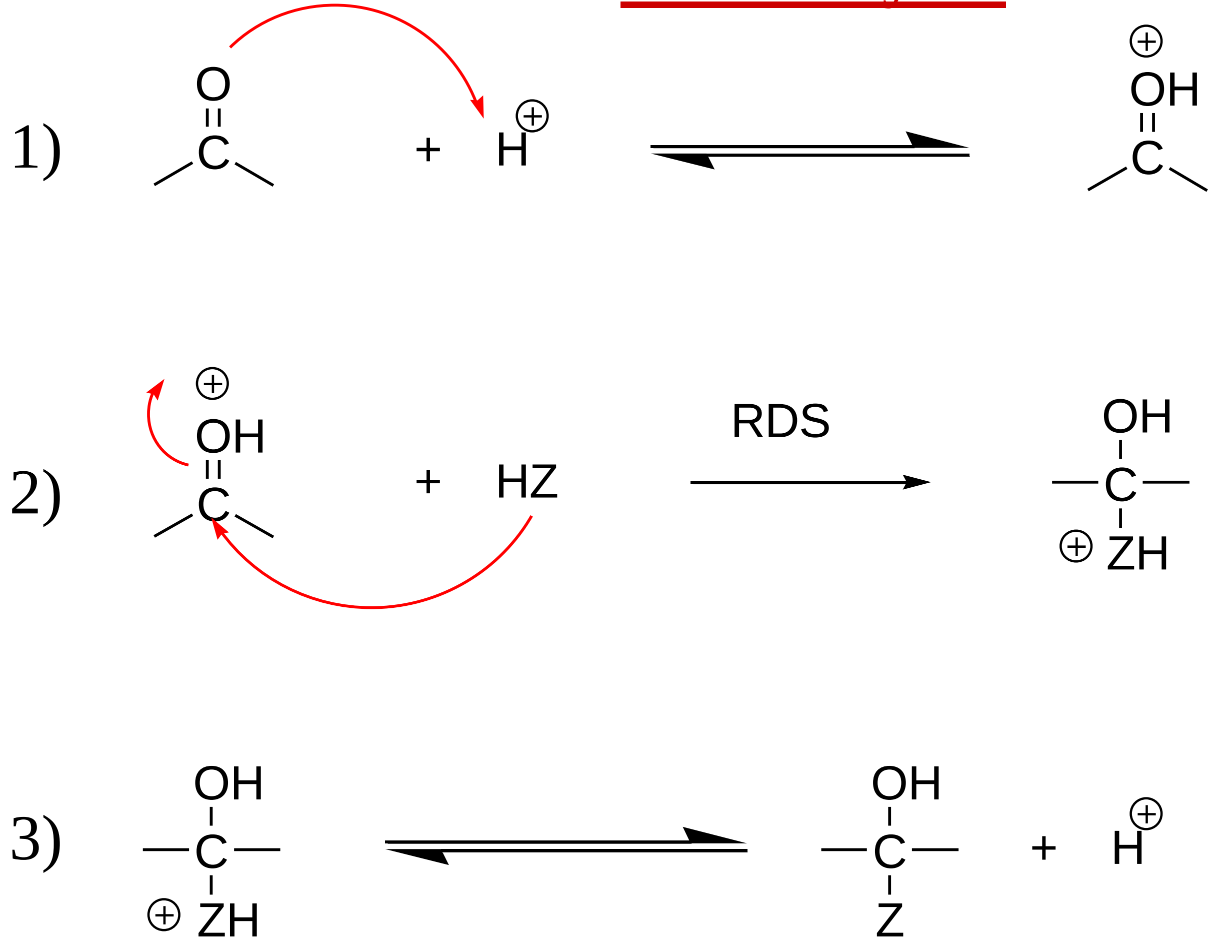


2)



Mechanism: nucleophilic addition to carbonyl,

➢ acid catalyzed



Chemical reactions of Aldehydes and ketones:

1) Oxidation
2) Reduction
3) Addition of cyanide
4) Addition of derivatives of ammonia
5) Addition of alcohols
6) Cannizzaro reaction
7) Addition of Grignard reagents

1) Oxidation

a) Aldehydes (**very easily oxidized**!)

$CH_3CH_2CH_2CH{=}O \quad + \quad KMnO_4$, etc. $\rightarrow \quad CH_3CH_2CH_2COOH$

carboxylic acid

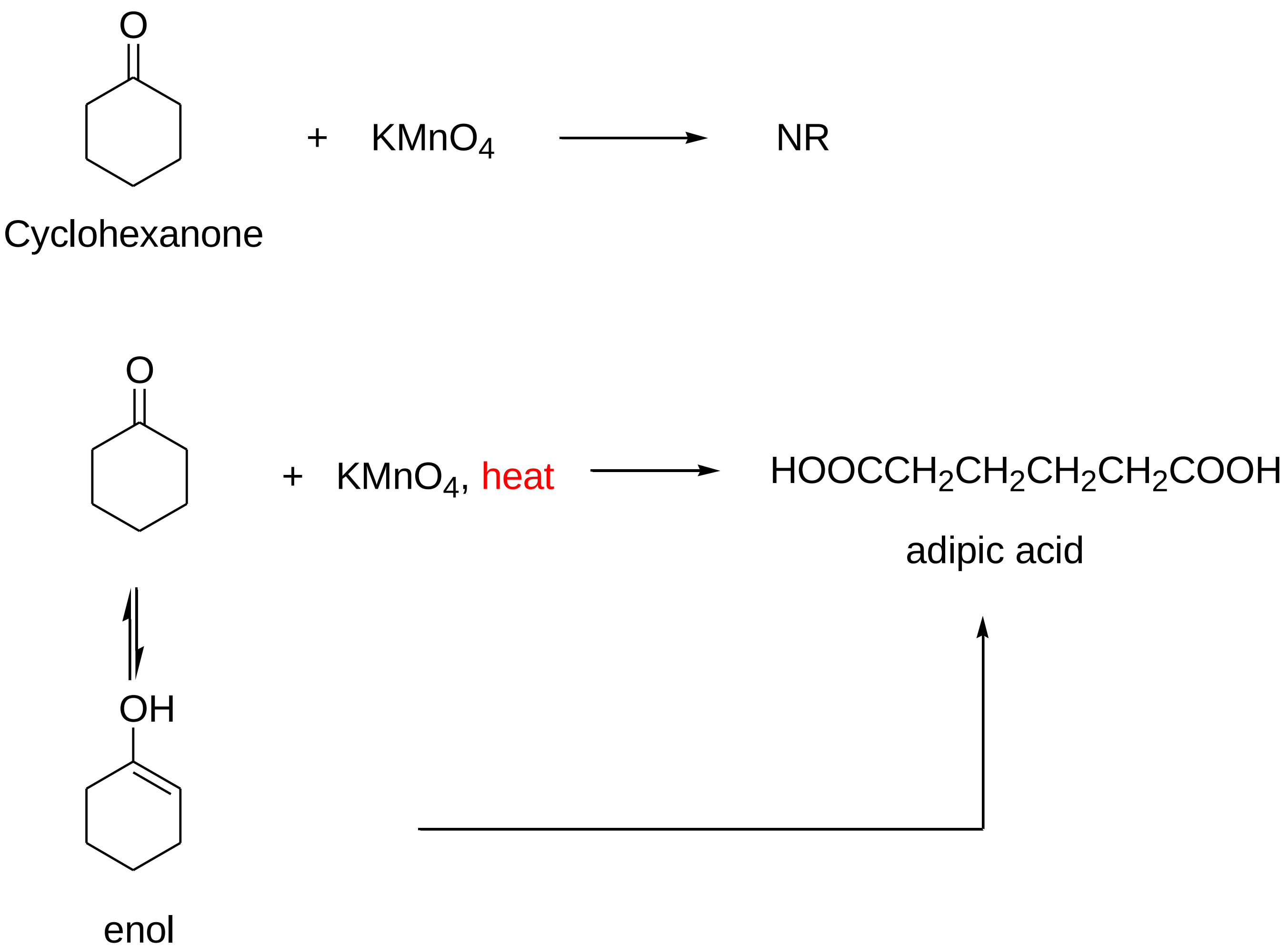
$CH_3CH_2CH_2CH{=}O \quad + \quad Ag^+ \quad \rightarrow \quad CH_3CH_2CH_2COO^- \quad + \quad Ag$

Silver mirror

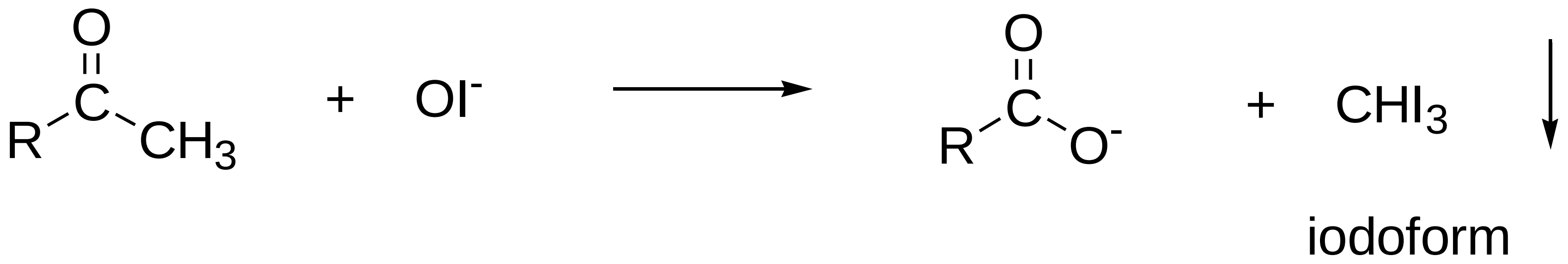
Tollen's test for easily oxidized compounds like aldehydes.

($AgNO_3$, $NH_4OH(aq)$)

Ketones only oxidize under <u>vigorous</u> conditions via the enol.



b) Methyl ketones:



Yellow ppt

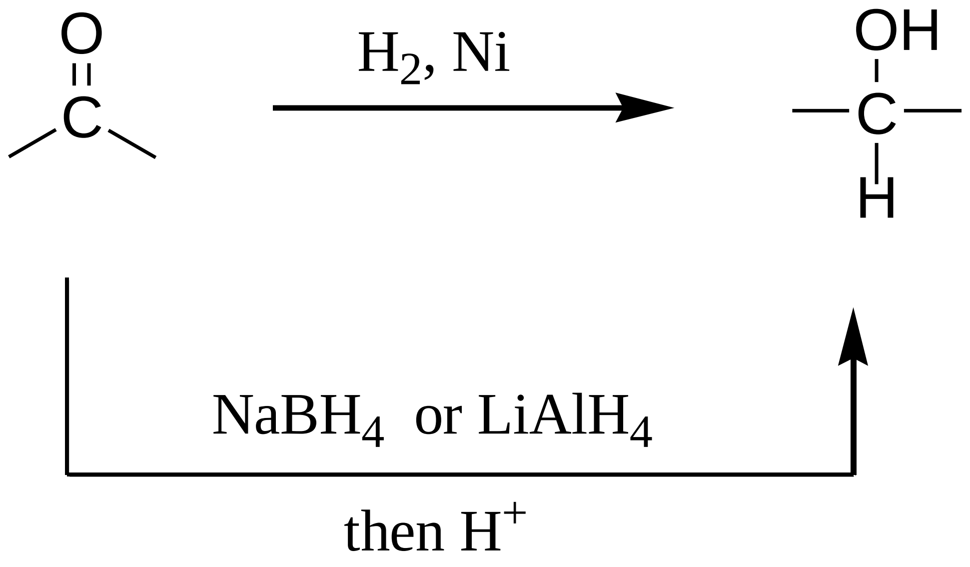
test for <u>methyl</u> ketones

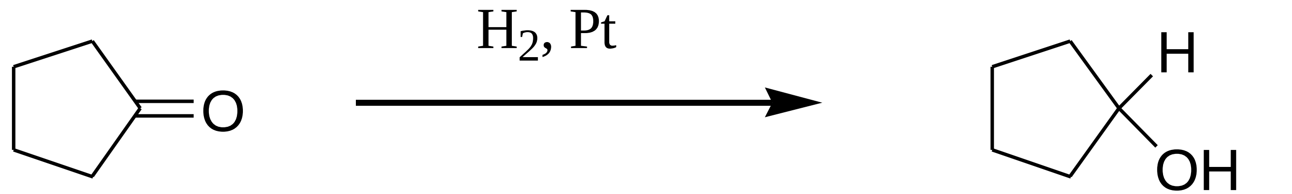
$$CH_3CH_2CH_2\overset{O}{\overset{\|}{C}}CH_3 + \text{(xs) NaOI} \longrightarrow CH_3CH_2CH_2CO_2^- + CHI_3$$

2-pentanone

2) Reduction:

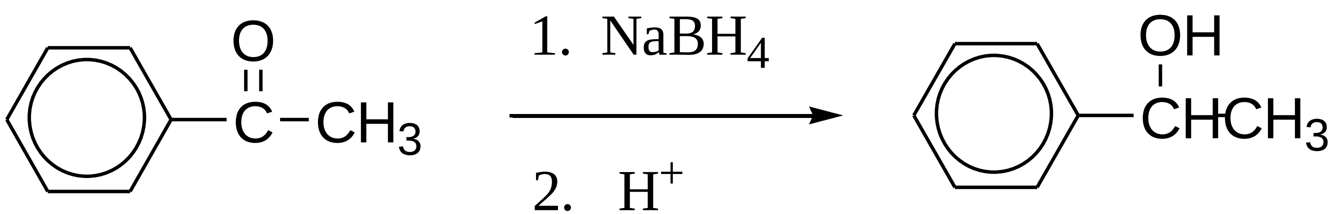
a) To alcohols





cyclopentanone

cyclopentanol

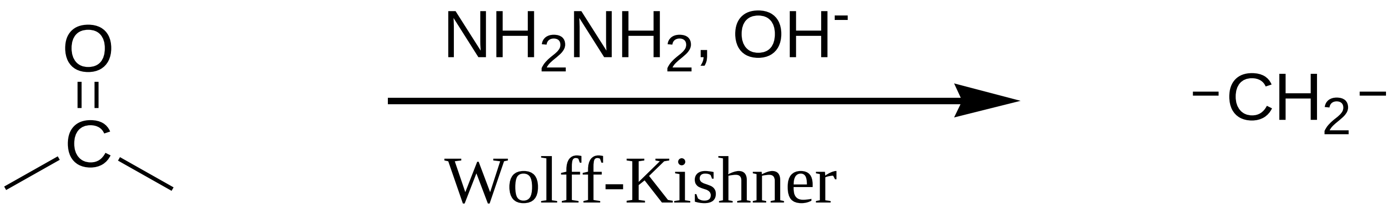


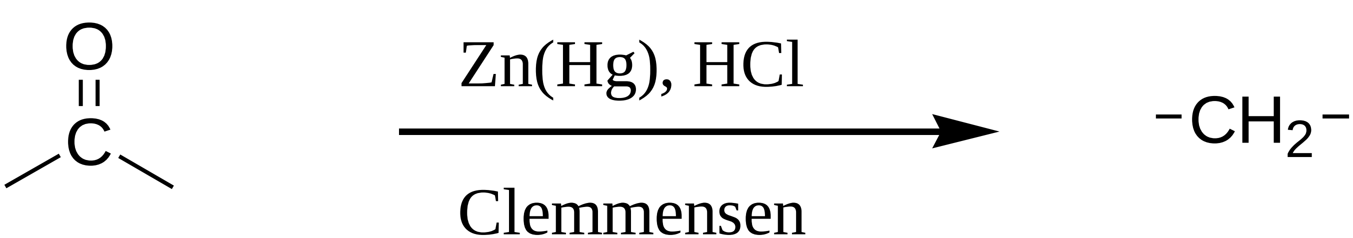
acetophenone

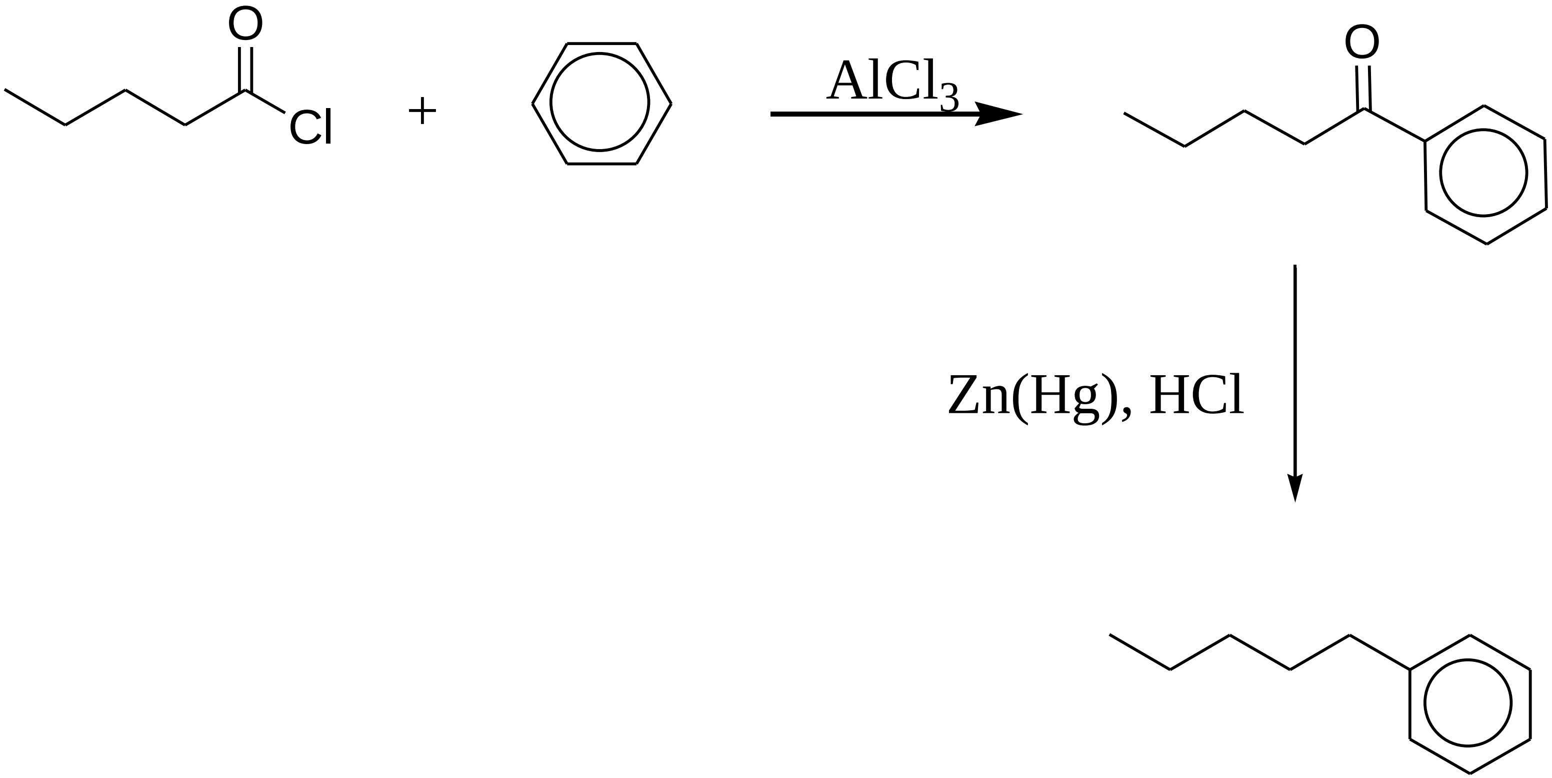
1-phenylethanol

Reduction

b) To hydrocarbons



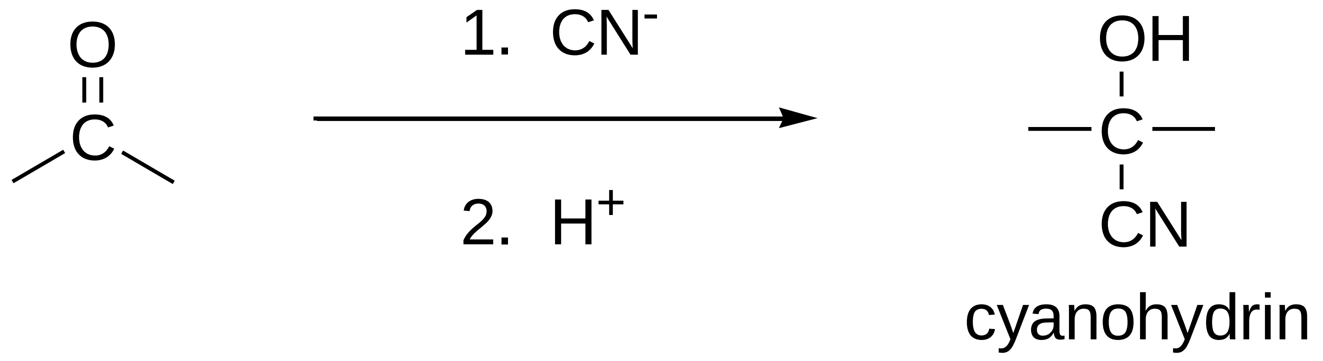




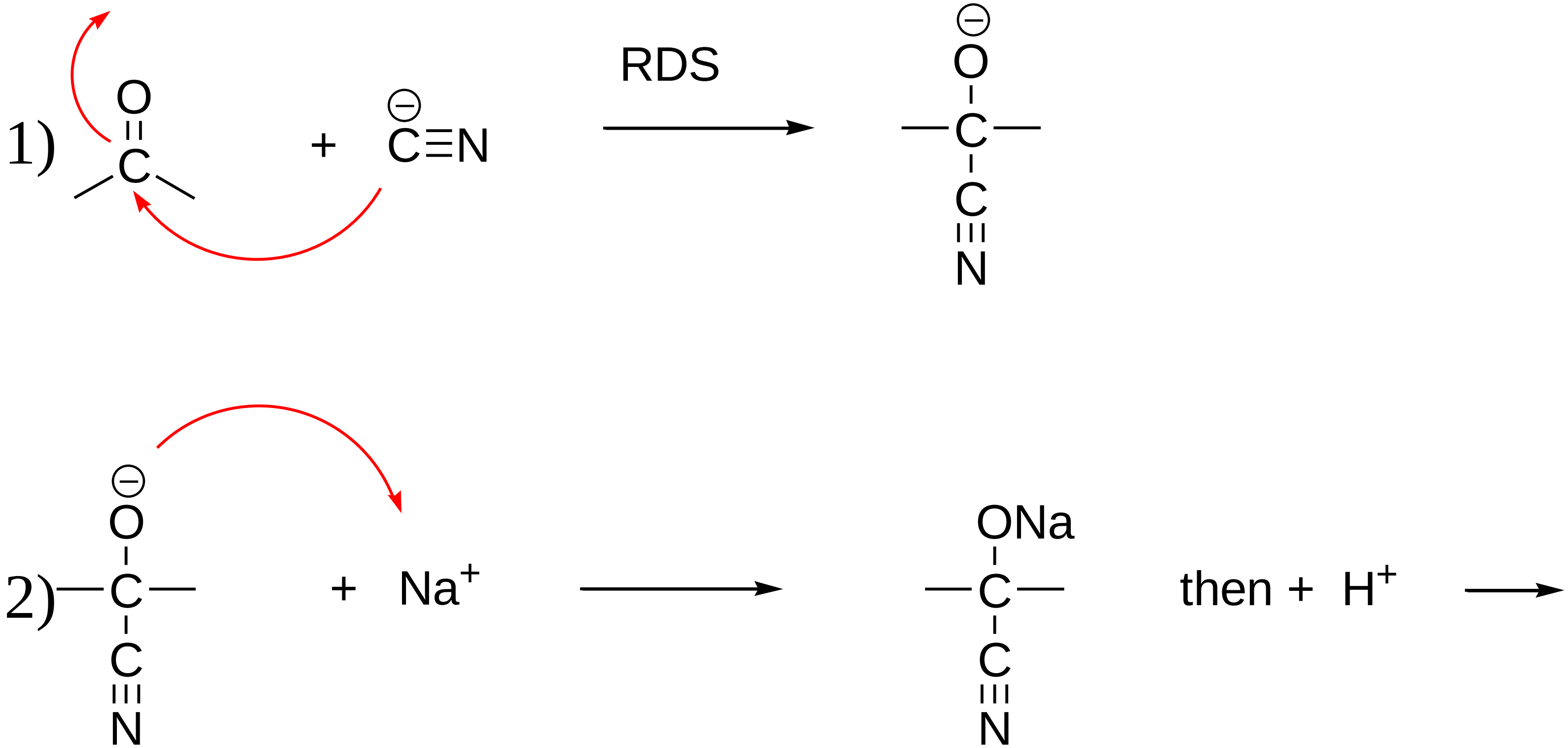
n-pentylbenzene

cannot be made by Friedel-Crafts alkylation
due to rearrangement of carbocation

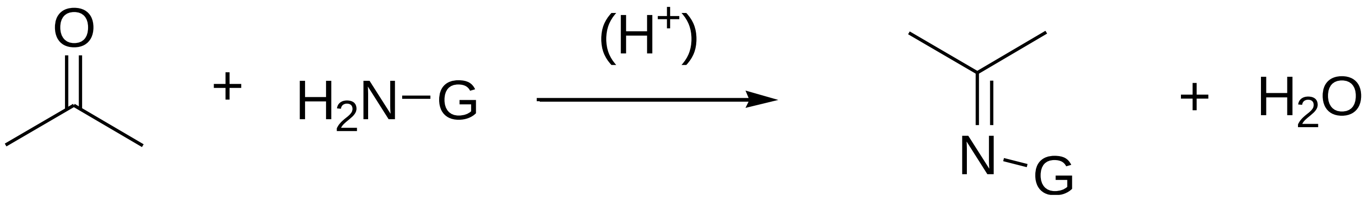
3) Addition of cyanide



mechanism for addition of cyanide
nucleophilic addition



4) Addition of derivatives of ammonia

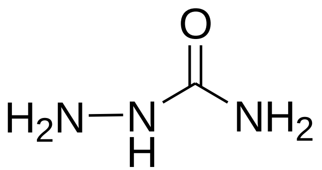


H_2N-NH_2

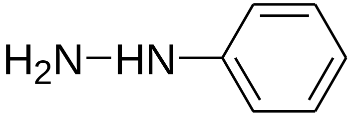
hydrazine

H_2N-OH

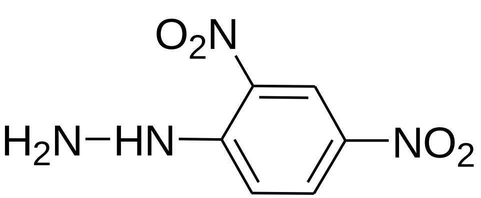
hydroxylamine



semicarbazide

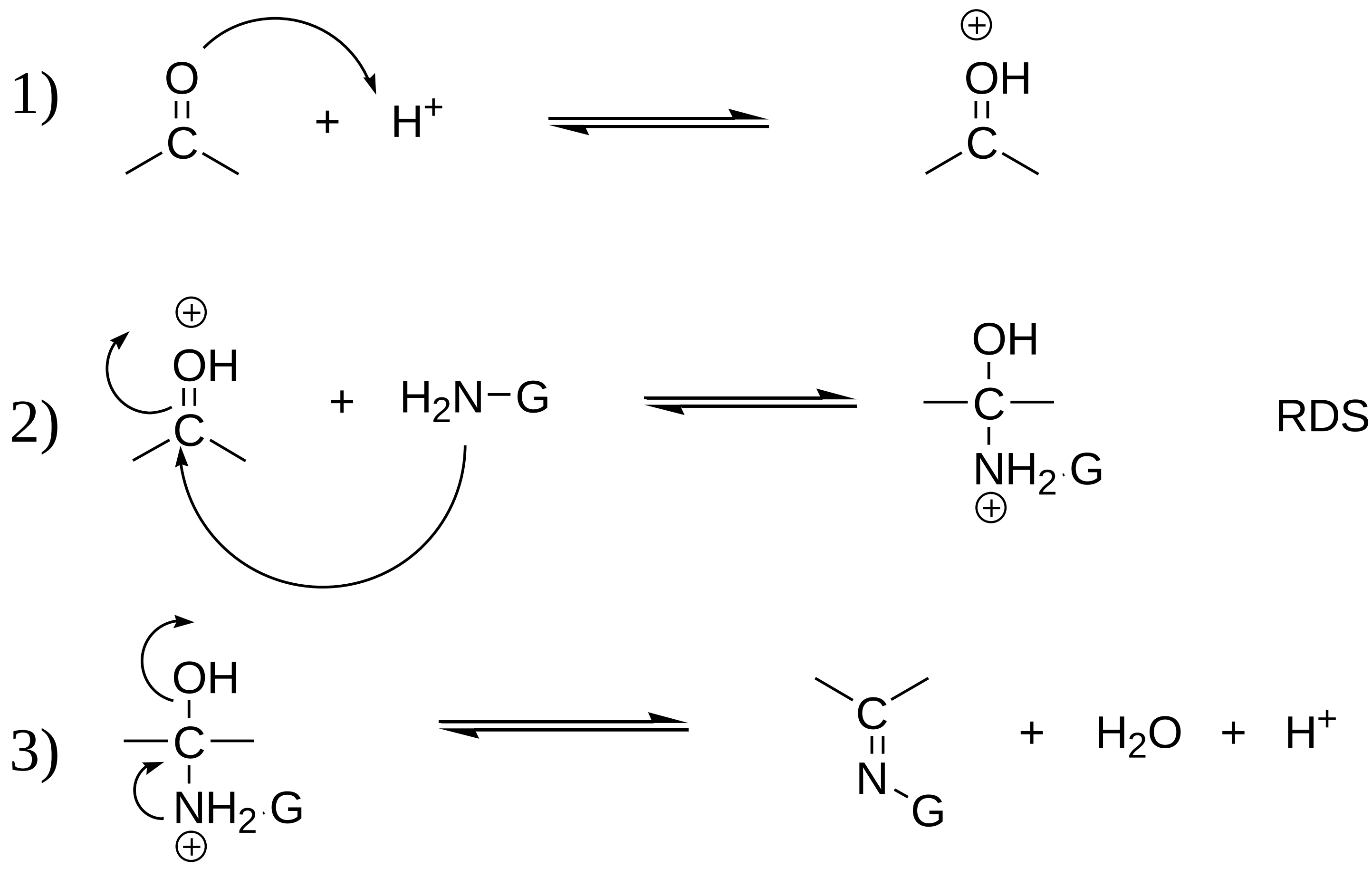


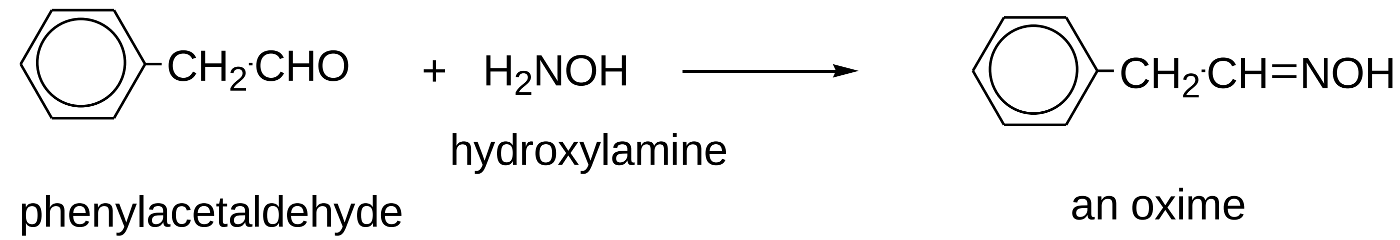
phenylhydrazine

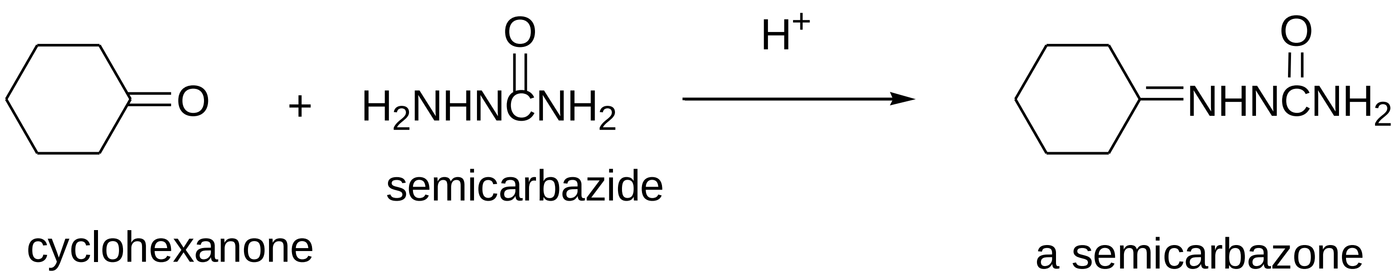


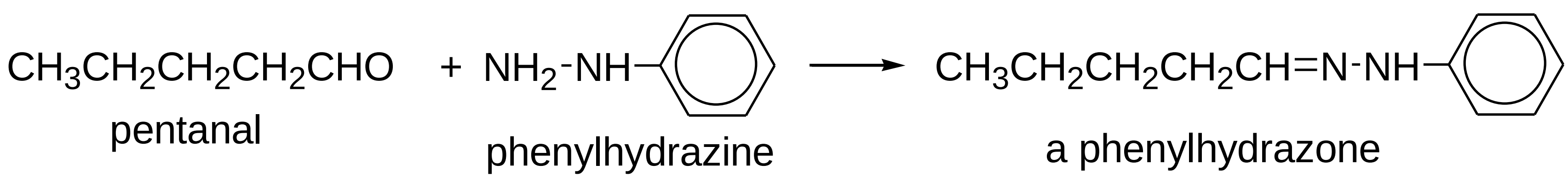
2,4-dinitrophenylhydrazine

acid catalyzed nucleophilic addition mechanism followed by dehydration

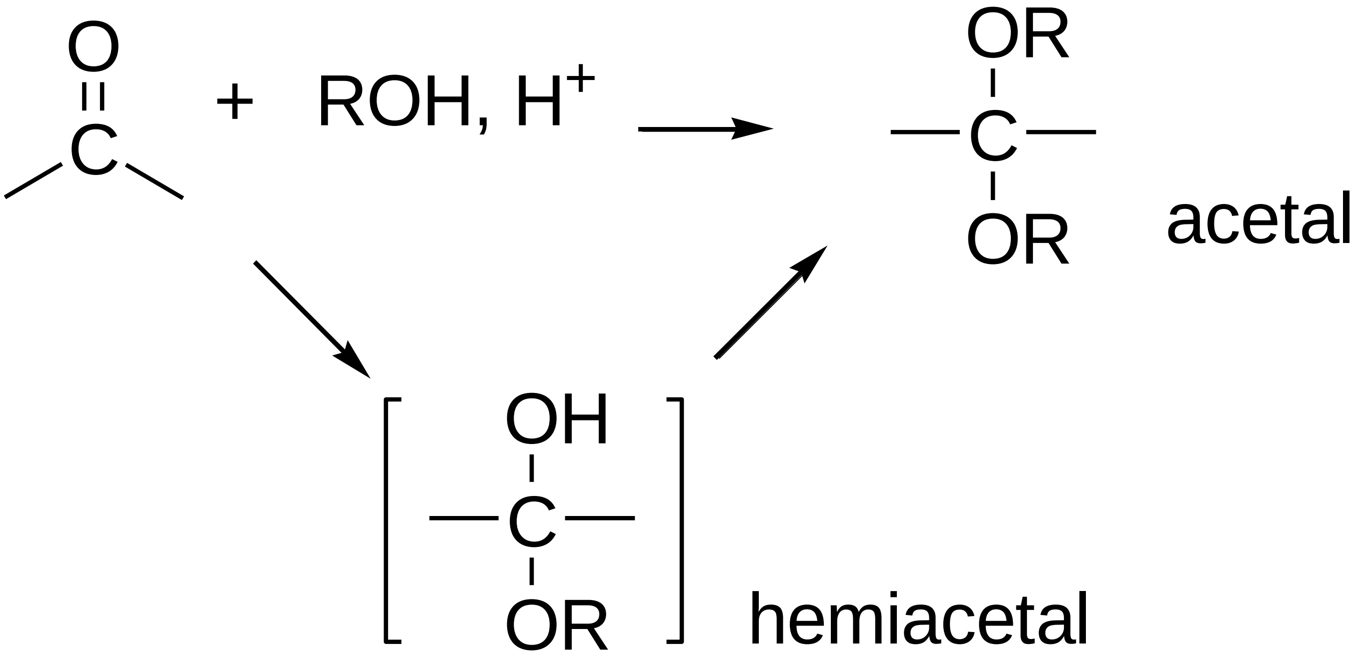



CH2CHO
+ H2NOH
hydroxylamine
phenylacetaldehyde
CH2CH=NOH
an oxime

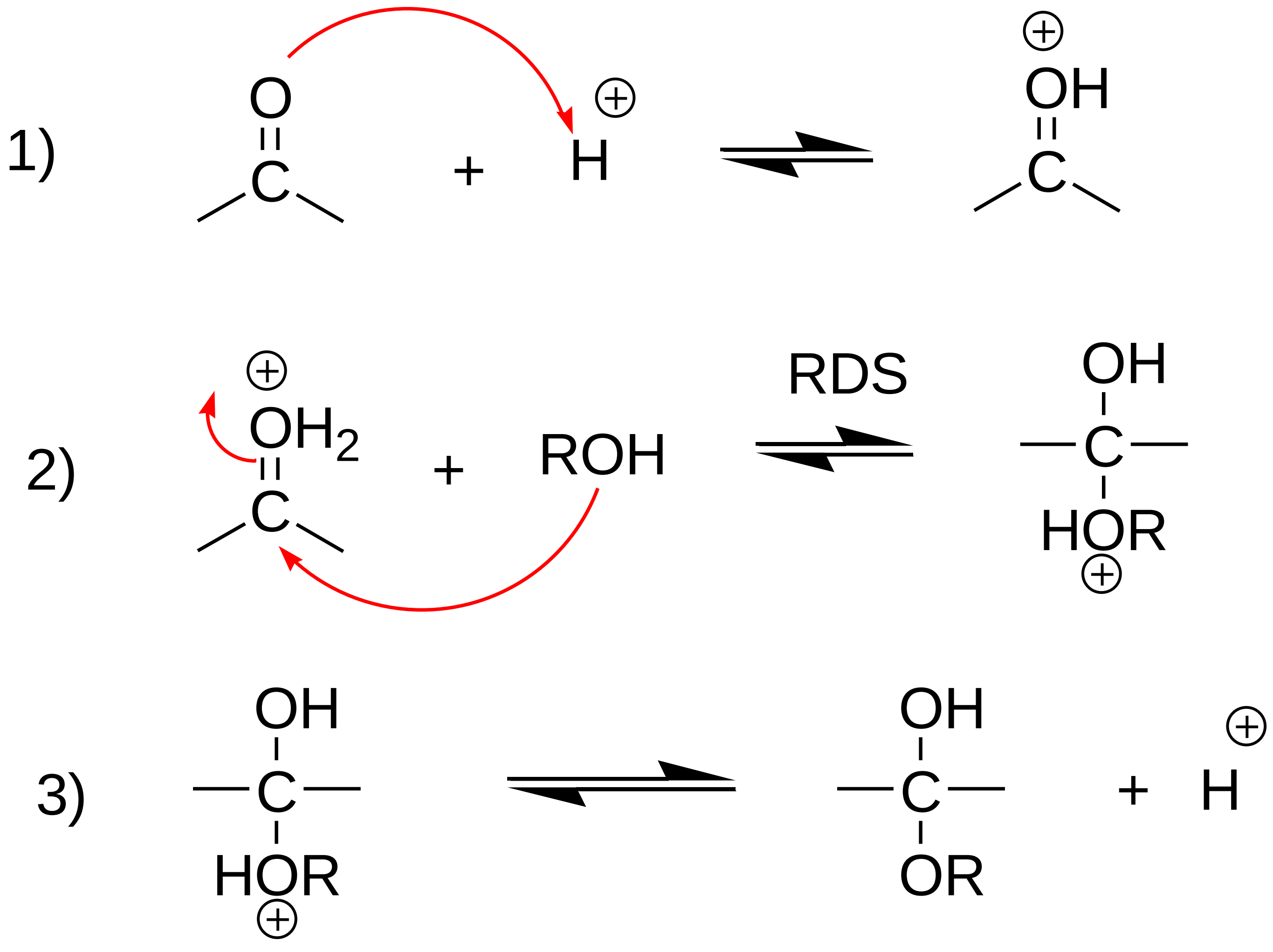

O
+ H2NHNCNH2
semicarbazide
cyclohexanone
H+
NHNCNH2
a semicarbazone

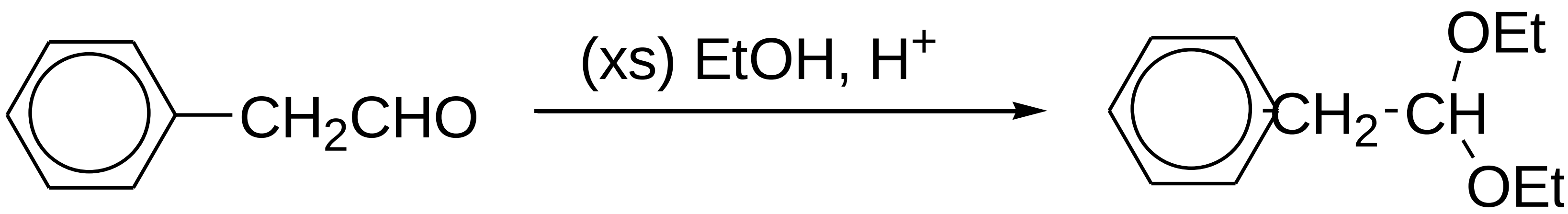

CH3CH2CH2CH2CHO
pentanal
+ NH2-NH
phenylhydrazine
CH3CH2CH2CH2CH=N-NH
a phenylhydrazone

5) Addition of alcohols

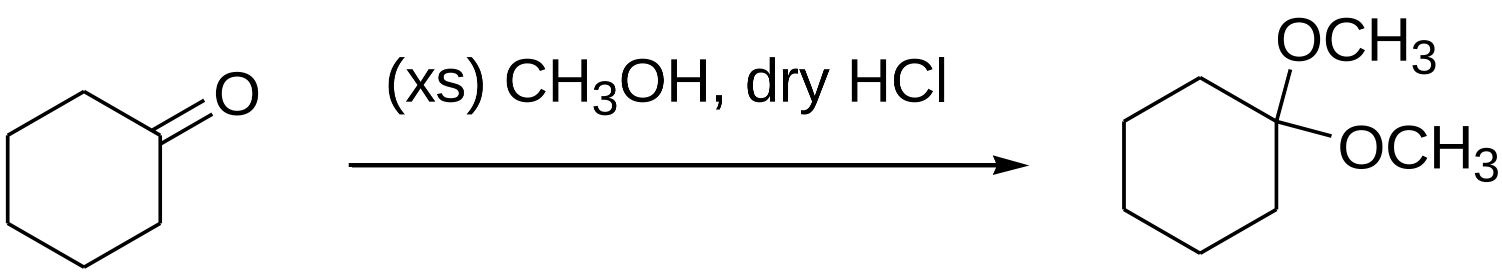


Mechanism = nucleophilic addition, acid catalyzed





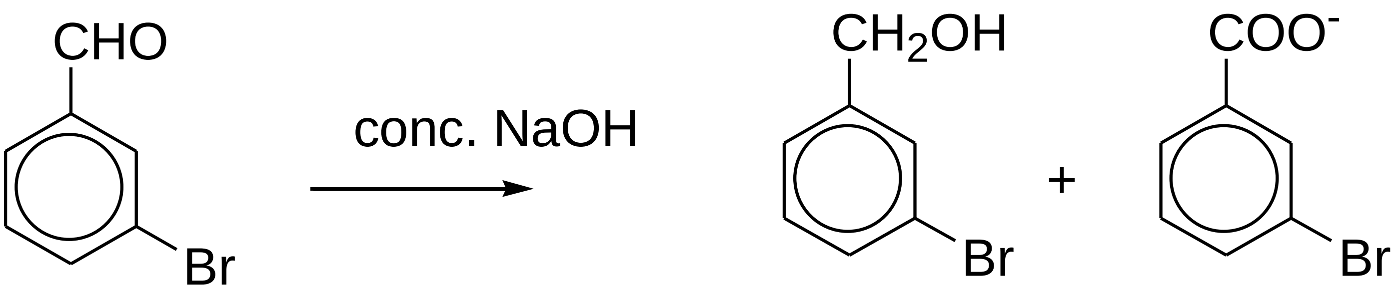
acetal



ketal

6) Cannizzaro reaction. (self oxidation/reduction)

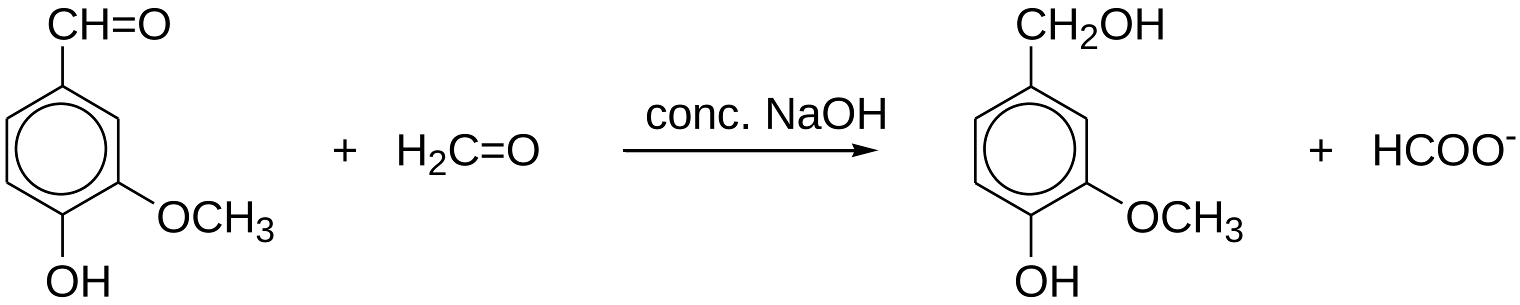
a reaction of **aldehydes without α-hydrogens**



$$H_2C{=}O \xrightarrow{\text{conc. NaOH}} CH_3OH + HCOO^-$$

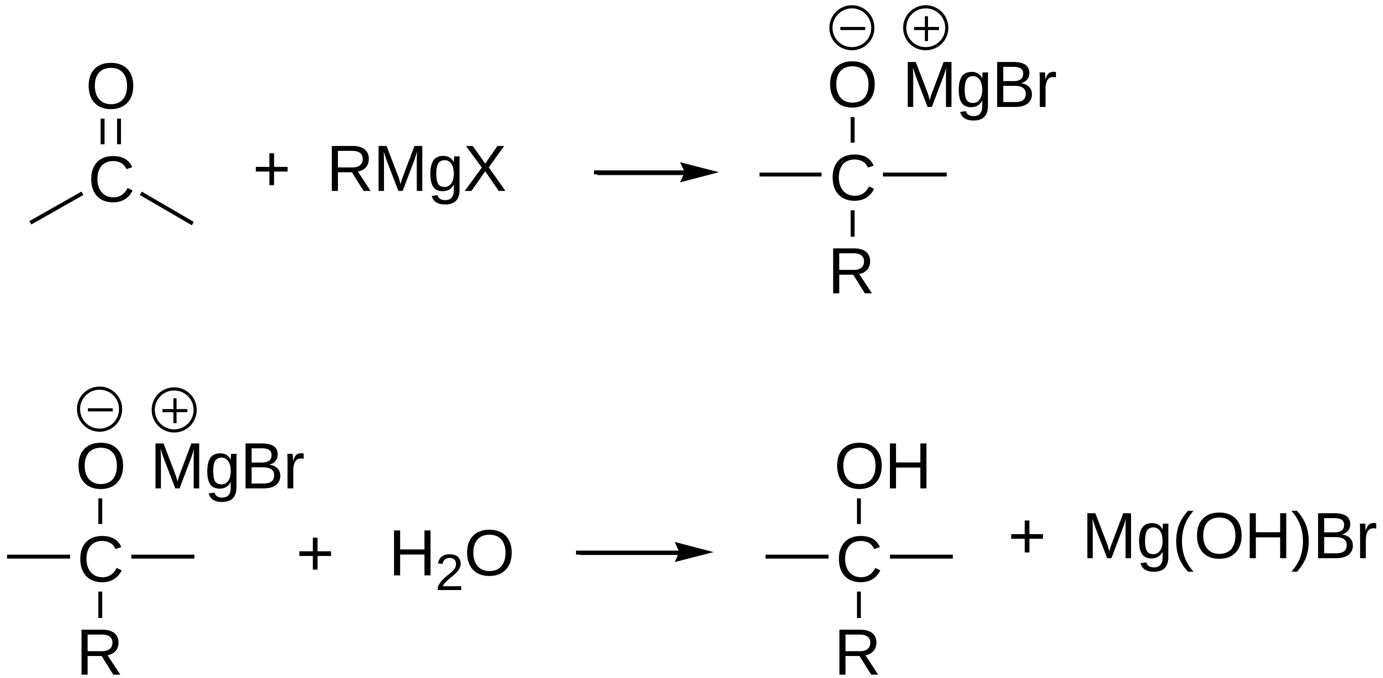
Formaldehyde is the most easily oxidized aldehyde. When mixed with another aldehyde that doesn't have any alpha-hydrogens and conc. NaOH, all of the formaldehyde is oxidized and all of the other aldehyde is reduced.

Crossed Cannizzaro:



vanillin

7) Addition of Grignard reagents.

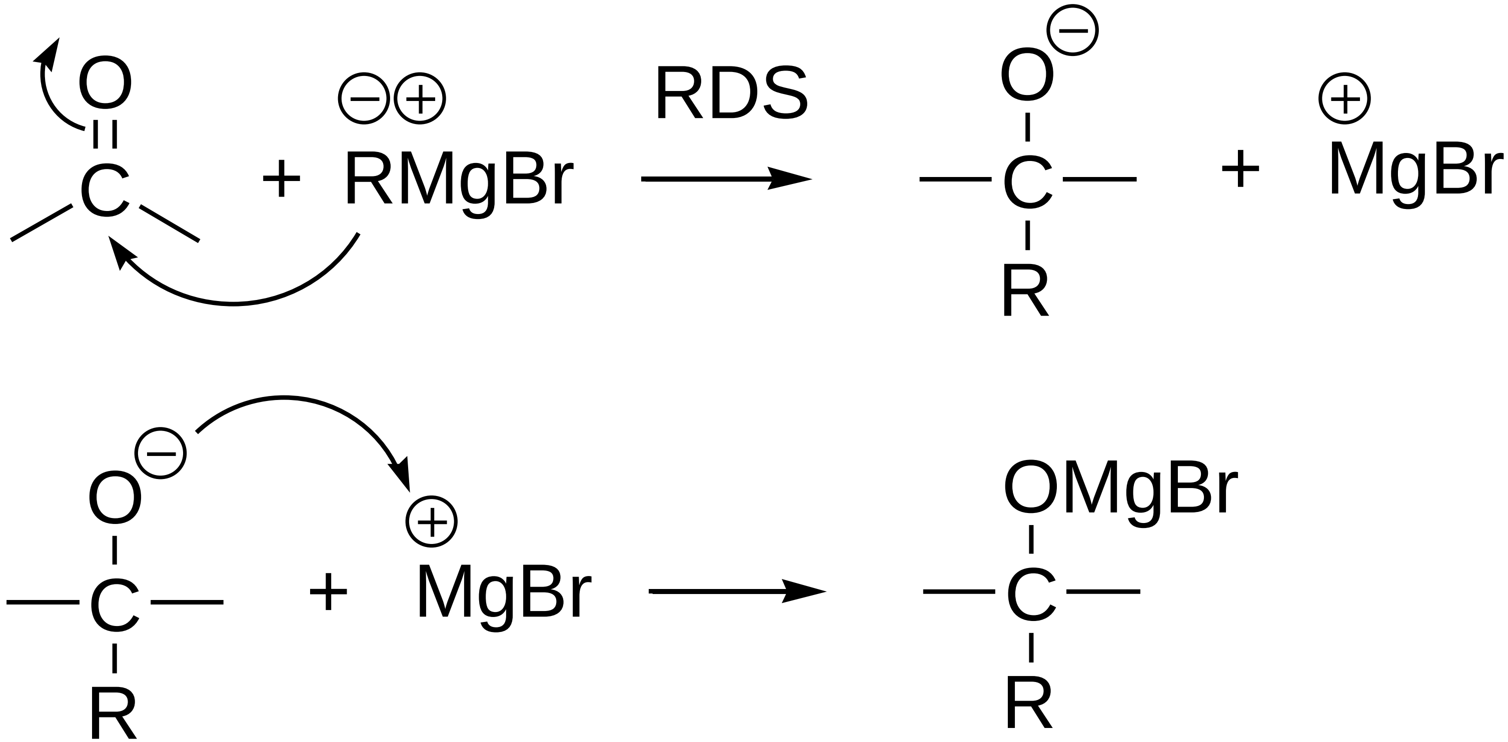


larger alcohol

mechanism = nucleophilic addition

1)

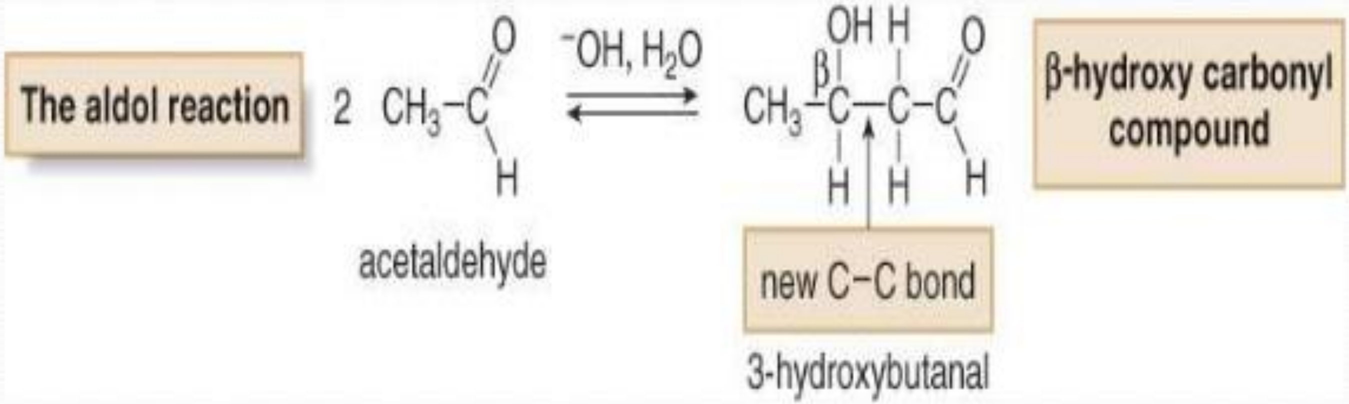
2)



ALDOL CONDENSATION

Condensation between two molecules of an Aldehyde or a ketone to form a β-hydroxyaldehyde or a β-hydroxy ketone is known as a ALDOL CONDENSATION.

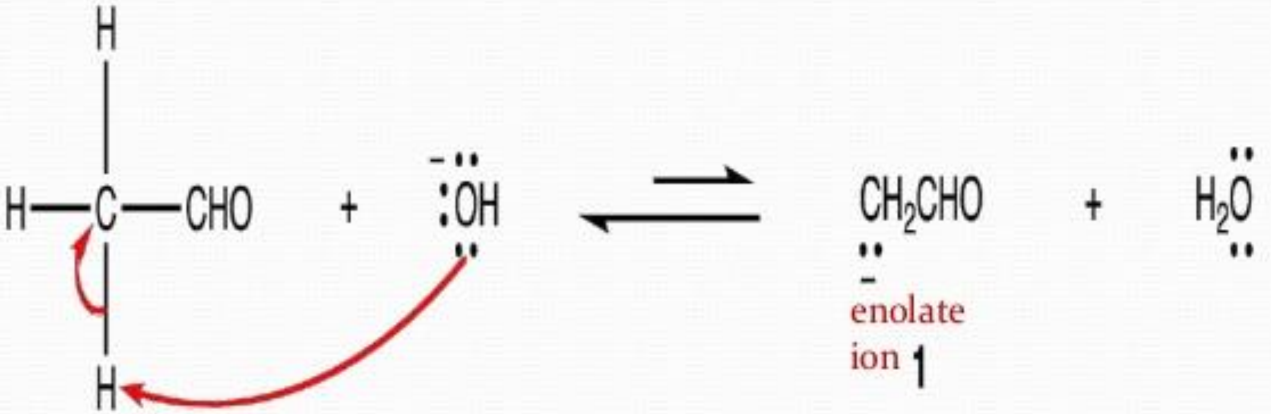
ALDOL CONDENSATION is possible only when the carbonyl compound contains atleast one α-hydrogen atom.



Thus the following Aldehydes or ketones having no α-hydrogen atom do not undergo Aldol Condensation.

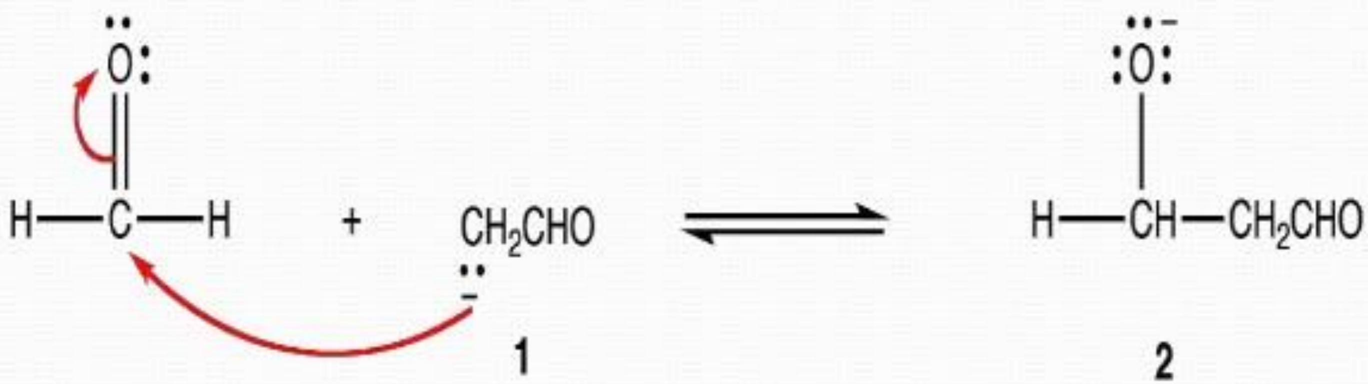
- **Step 1:**
  First, an acid-base reaction. Hydroxide functions as a base and removes the acidic a-hydrogen giving the reactive enolate.
  Step 2:
  The nucleophilic enolate attacks the aldehyde at the electrophilic carbonyl **C** in a nucleophilic addition type process giving an intermediate alkoxide.
  Step 3:
  An acid-base reaction. The alkoxide deprotonates a water molecule creating hydroxide and the β-hydroxyaldehydes or **aldol** product.

Step:1



❖An enolate ion is the anion formed when an alpha hydrogen in the molecule of an aldehyde or a ketone is removed as a hydrogen ion.

Step:2



The alkoxide ion is the conjugate base of alcohols.

Step:3

2

3

aldol product

Alkoxide ion is protonated by water.

Knoevenagel reaction

Introduction

- Knoevenagel condensation is nucleophilic addition of an active hydrogen compound to a carbonyl group followed by a dehydration reaction in which a molecule of water is eliminated (hence Condensation). The product is often an alpha, beta conjugated enone.

- Knoevenagel reaction is a modification of Aldol condensation reaction.

- The active hydrogen compound used should be of the form $Z\text{-}CH_{2\text{-}}Z$ or Z-CHR-Z where Z is an electron withdrawing group.

- The carbonyl group is an aldehyde or a ketone.
- Eg: Acetaldehyde (CH_3CHO), Acetone ($CH_3CO\ CH_{3)}$
- The catalyst is usually a weakly basic amine.
- Eg: Pyridine (C_5H_5N)

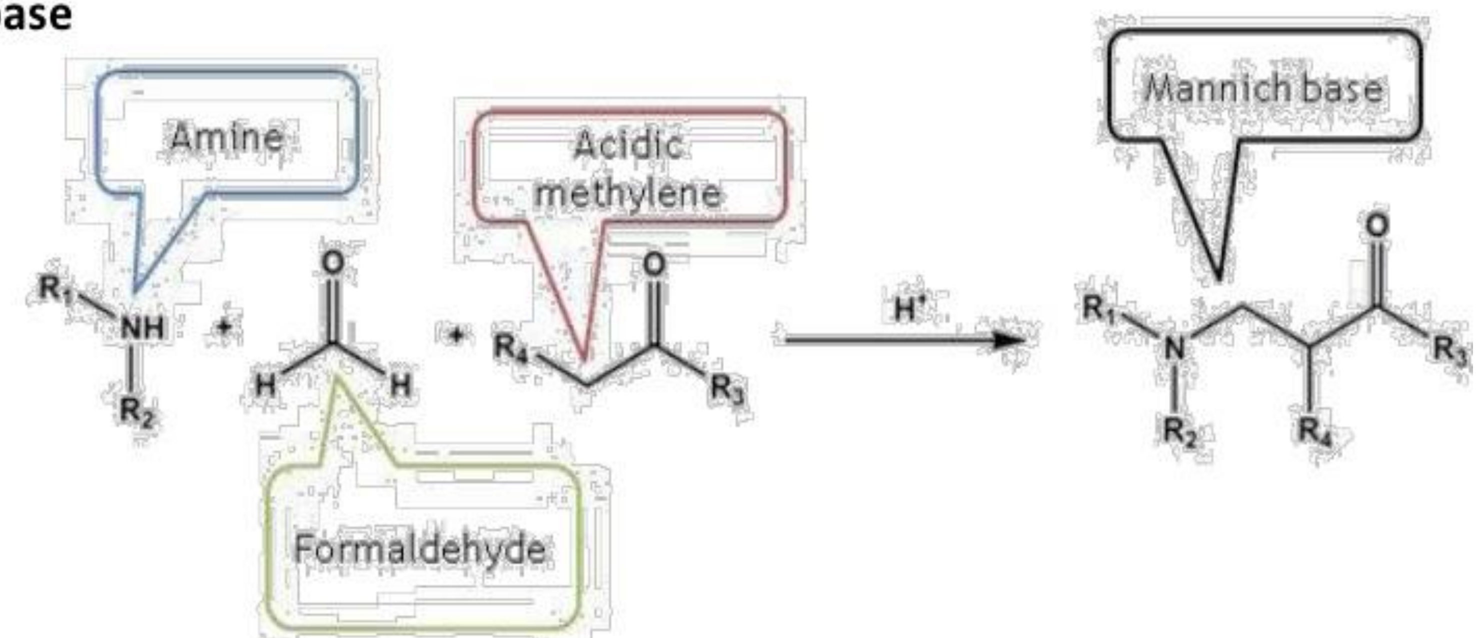
Knoevenagel reaction

$$R\text{-}CHO + CH_2(COOR)_2 \xrightarrow{\text{base}} RCH{=}C(COOR)_2 \xrightarrow[CO_2]{OH^-} RCH{=}CH\text{-}COOH$$

Knoevenagel condensation is nucleophilic addition of an active hydrogen compound to a carbonyl group followed by a dehydration reaction in which a molecule of water is eliminated (hence Condensation). The product is often an alpha, beta conjugated enone.

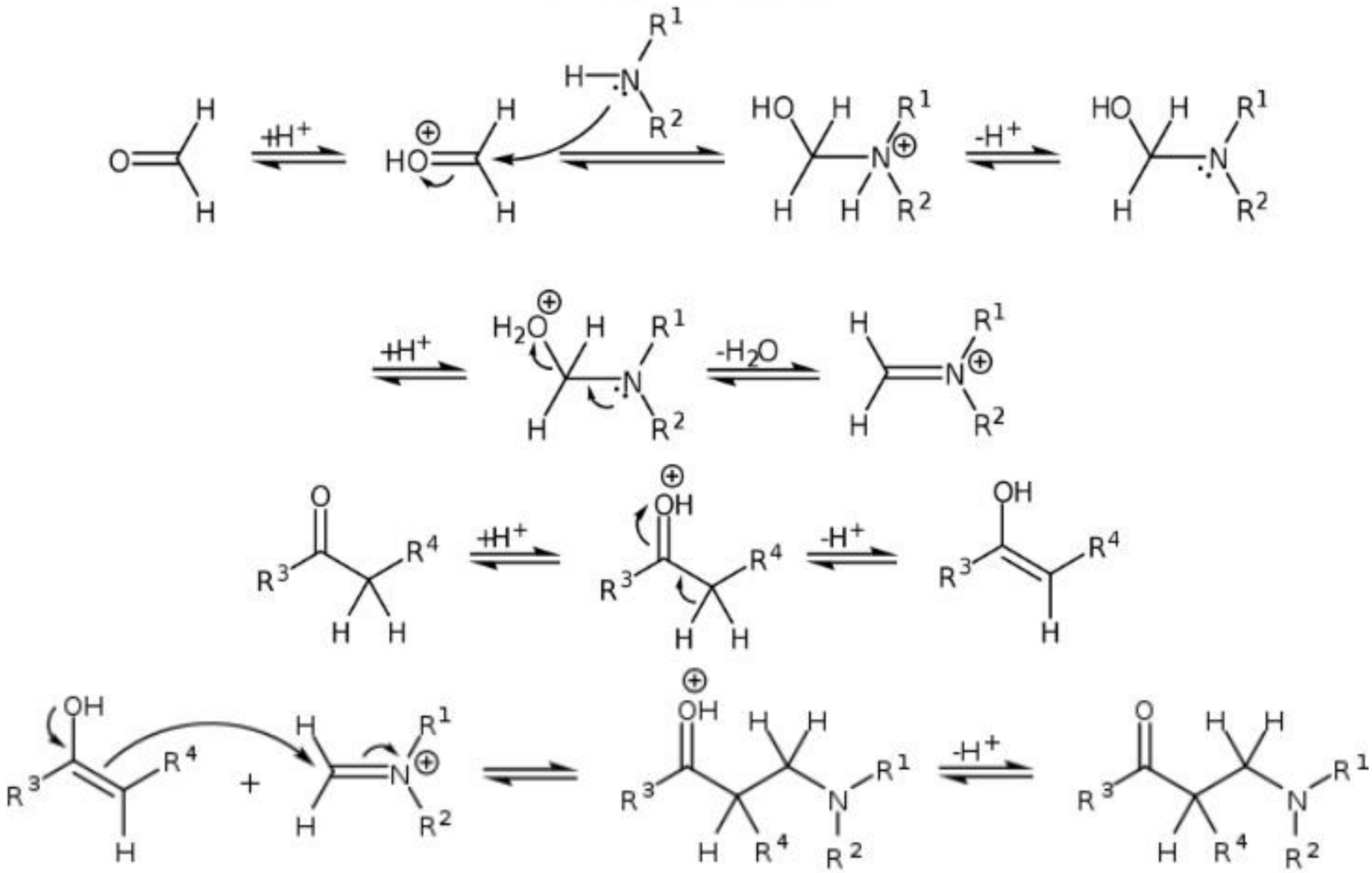
Background

The **Mannich reaction** is the aminoalkylation reaction, involving the condensation of an enolizable carbonyl compound with a nonenolizable aldehyde (like formaldehyde) and ammonia, or a primary or a secondary amine to furnish a β-aminocarbonyl compound, also known as **Mannich base**



Li, J.J. *Name Reactions.* 2. Springer **2003**.

Mechanism



Li, J.J. *Name Reactions.* 2. Springer **2003**.