# SPLICING SYSTEMS

## ACCEPTING VS. GENERATING


Juan Castellanos

Victor Mitrana

Eugenio Santos



Polytechnic University of Madrid

## OUTLINE

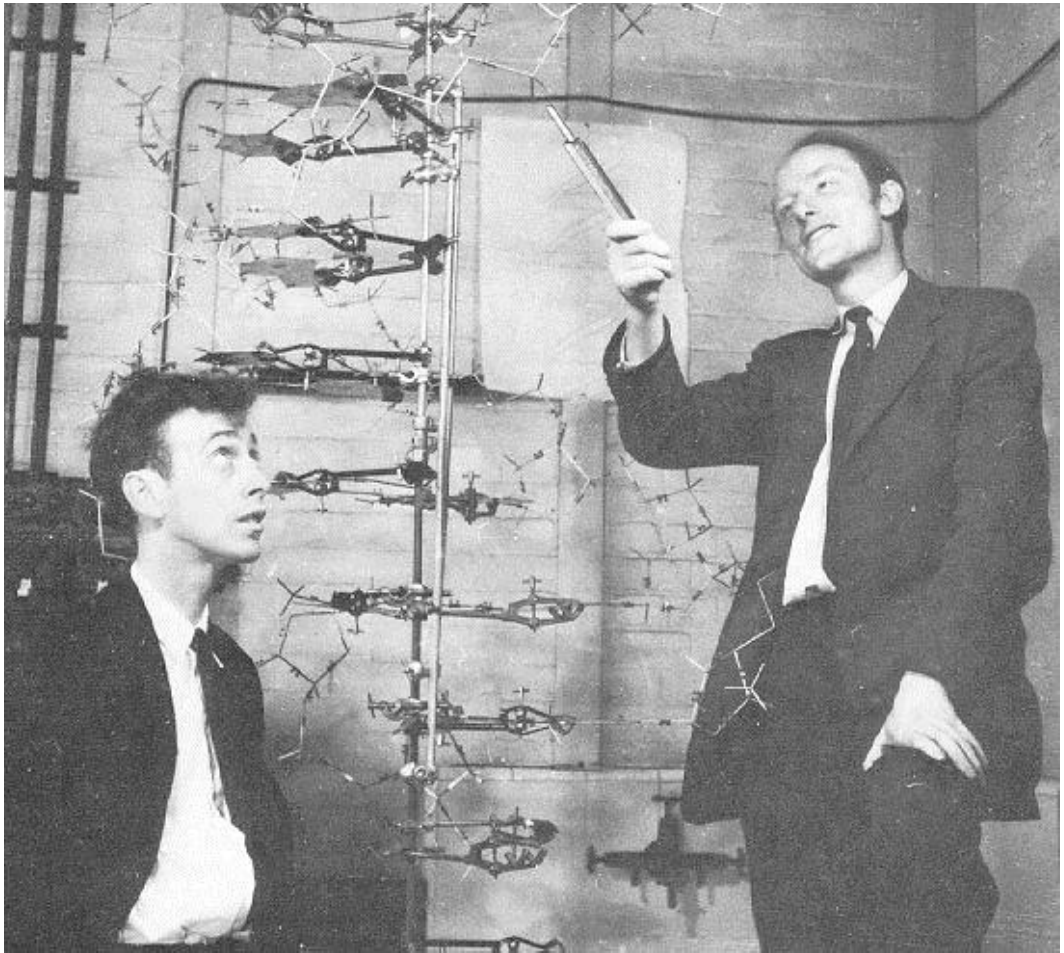
Structure and recombination of DNA
Splicing operation and splicing systems
Generating variants
Accepting variants
Comparison between the two variants
Properties of the accepting variants
Conclusion

# DNA (deoxyribonucleic acid)

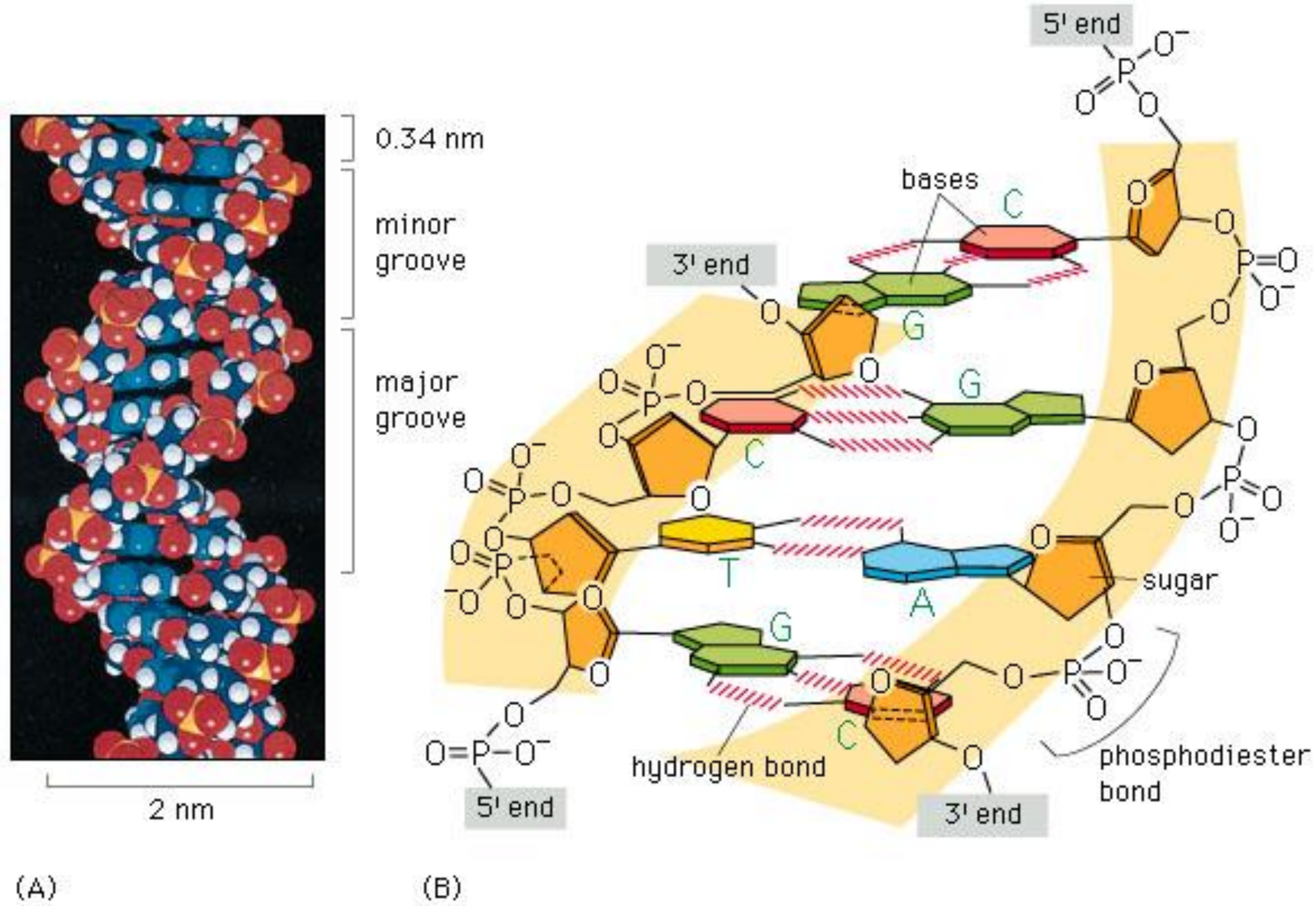
Watson & Crick (1953): *Nature* 25: 737-738

Molecular Structure of Nucleic Acids: a structure for deoxyribose nucleic acid.

Nobel Prize, 1962.



# DNA as computing tool (I)



# DNA as computing tool (II)

5' G→T→A→A→A→G→T→C→C→C→G→T→T→A→G→C 3'

# DNA as computing tool (III)

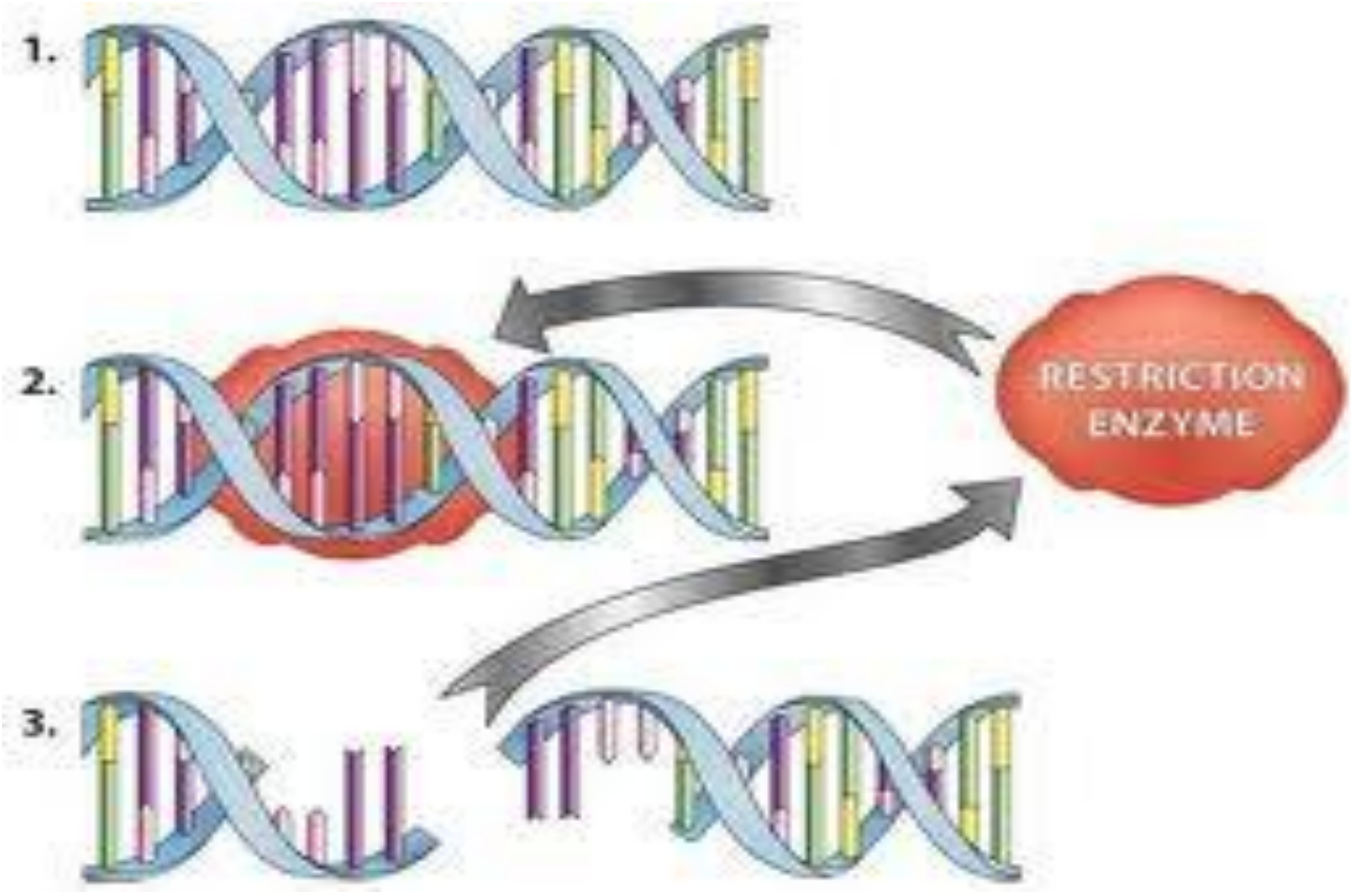
```
5' G→T→A→A→A→G→T→C→C→C→G→T→T→A→G→C 3'
   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
3' C←A←T←T←T←C←A←G←G←G←C←A←A←T←C←G 5'
```

# DNA as computing tool (IV)

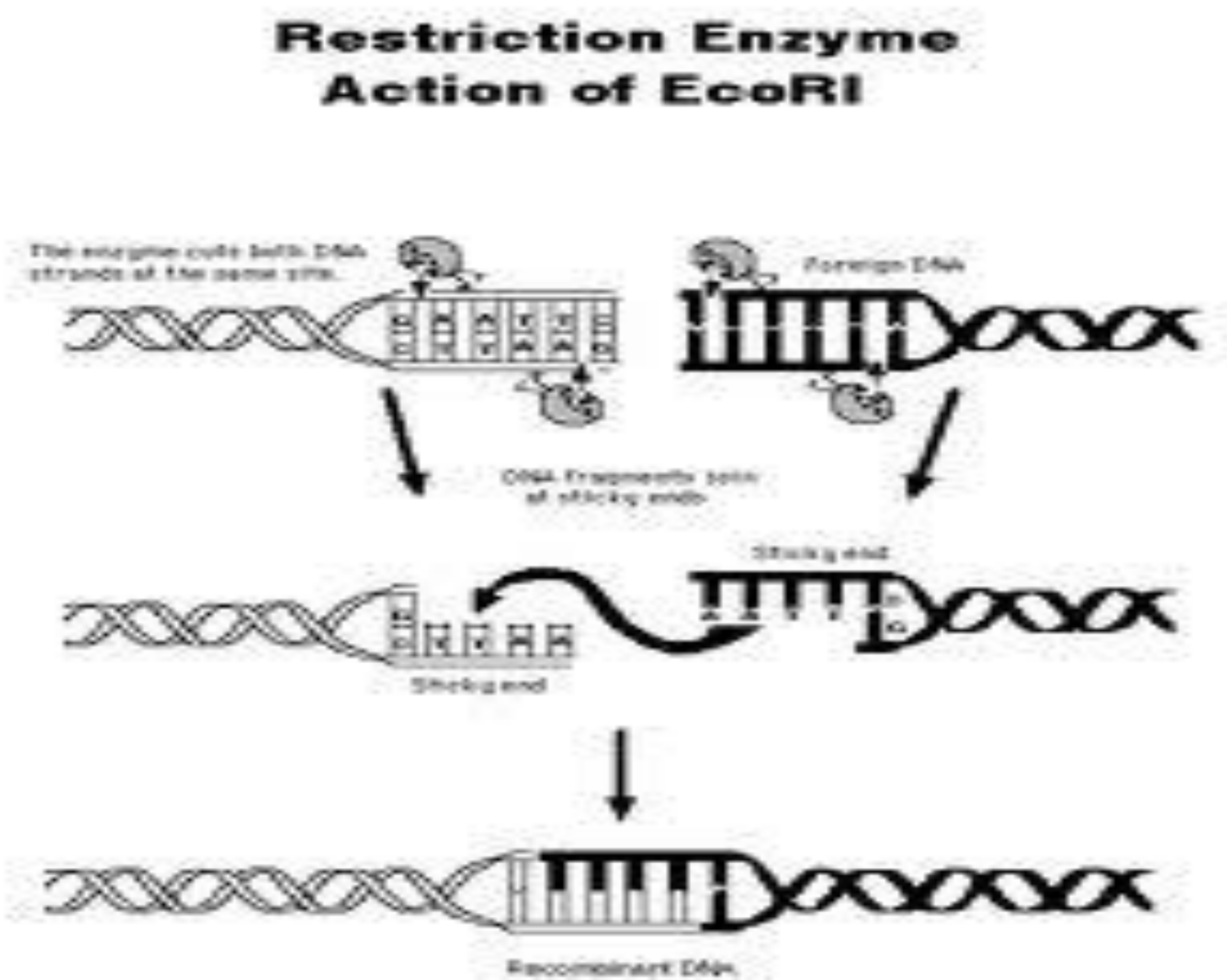
Circular DNA and Splicing Systems

DNA molecules exist not only in linear forms but also in circular forms.

# DNA Recombination: Enzymes (I)



# DNA Recombination: Enzymes (II)



# Tom Head

Department of Mathematical Sciences
Binghamton University

Areas of interest

Algebra

Computing with biomolecules

Formal representations of communication

http://www.math.binghamton.edu/tom/

# Splicing in nature (Head)

| | | |
|---|---|---|
| 5'-CCCCCTCGACCCCC-3' | TCGA | *Taq*I |
| 3'-GGGGGAGCTGGGGG-5' | AGCT | |
| 5'-AAAAAGCGCAAAAA-3' | GCGC | *Sci*NI |
| 3'-TTTTTCGCGTTTTT-5' | CGCG | |
| 5'-TTTTTGCGCTTTTT-3' | GCGC | *Hha*I |
| 3'-AAAAACGCGAAAAA-5' | CGCG | |

5'-CCCCCTCGCAAAAA-3'
3'-GGGGGAGCGTTTTT-5'

5'-AAAAAGCGACCCCC-3'
3'-TTTTTCGCTGGGGG-5'

(u,x,v):
$(T,CG,A) \in \mathbf{R}_1$
$(G,CG,C) \in \mathbf{R}_1$
$(C,GC,C) \in \mathbf{R}_2$

# Formal splicing (Head)

Given $w_1$, $w_2$ and $(u_1,x_1,v_1),(u_2,x_2,v_2) \in \mathbf{R}_1$

$w_1 = w'_1 u_1 x_1 v_1 w''_1$ $\qquad$ $w_2 = w'_2 u_2 x_2 v_2 w''_2$

$z_1 = w'_1 u_1 x v_2 w''_2$ $\qquad$ $z_2 = w'_2 u_2 x v_1 w''_2$

*provided* $x = x_1 = x_2$.

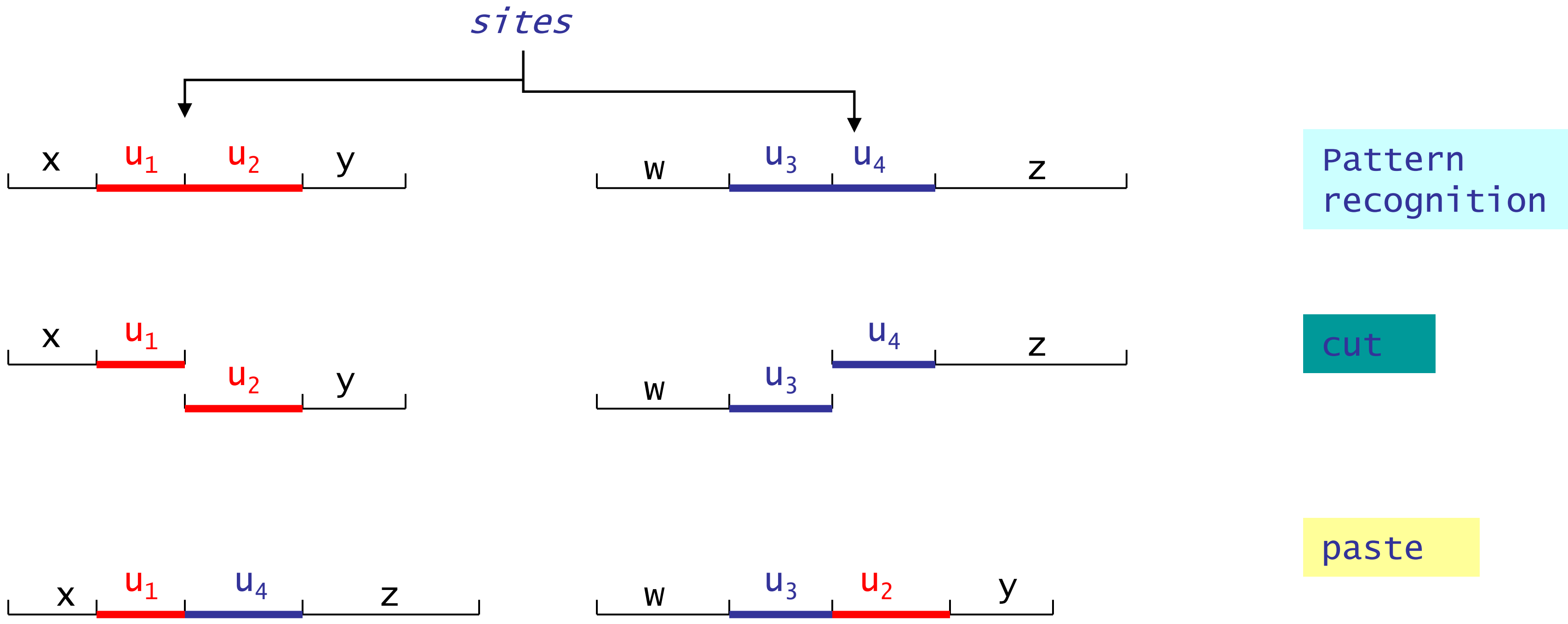
Further formalization: $((u_1,x,v_1),(u_2,x,v_2)) \in \mathbf{R}_1$

Still: $((u_1 x,v_1), (u_2 x,v_2)) \in \mathbf{R}_1$

# Formal splicing (II)

$r = u_1\#u_2 \ \$ \ u_3\#u_4$ rule

$\sigma$: $(x\ u_1u_2\ y,\ \ wu_3u_4\ z) \longrightarrow (x\ u_1\ u_4\ z\ ,\ \ wu_3\ u_2\ y)$



# Formal splicing (III)

Splicing scheme (H scheme) [Head 1987]: $\sigma=(V,R)$
R may be infinite [Păun 1996]

$(x,y) \succ\!\succ_r z,w$ *iff* $x = x_1u_1u_2x_2$, $y = y_1u_3u_3y_2$
$z = x_1u_1u_3y_2$, $w = y_1u_3u_2x_2$

$\sigma(x,y)=\{z,w \mid (x,y) \succ\!\succ_r z,w, \quad r\in R\}$

$$\sigma(L)=\bigcup_{x,y\in L}\sigma(x,y)$$

Note: $\sigma(x,y)\neq \sigma(\{x,y\})$.

$\sigma^*(L)= \bigcup_{k\geq 0}\sigma^k(L)$, $\sigma^0(L)= L$, $\sigma^{k+1}(L)= \sigma^k(L) \cup \sigma(\sigma^k(L))$

# Formal splicing (IV)

- There is a solid theoretical foundation for splicing as an operation on formal languages.

- The basic model is a single tube, containing an initial population of dsDNA, several restriction enzymes, and a ligase. Mathematically this is represented as a set of strings (the initial language), a set of cutting and pasting operations.

- In biochemical terms, procedures based on splicing may have some advantages, since the DNA is used mostly in its double stranded form, and thus many problems of unintentional annealing may be avoided.

# Formal splicing – Extended H systems

$\gamma = (V, T, A, R)$

$V$ alphabet

$T \subseteq V$ terminal alphabet

$A \subseteq V^*$ set of strings

$R$ splicing rules

$L(\gamma) = \sigma^*(A) \cap T^*$

Theorem

1. [Culik II, Harju 1991], [Pixton 1996]
   $H(FIN, FIN) \subset REG$.
2. [Păun, Rozenberg, Salomaa 1996]
   $EH(FIN, FIN) = REG$.

# H-system with multiplicities

$M_1 \Rightarrow_\sigma M_2$ iff there are $x,y,z,w$ such that

(i) $M_1(x)>0$, $(M_1-(x,1))(y)>0$,
(ii) $z,w \in \sigma(x,y)$
(iii) $M_2 = M_1-(x,1)-(y,1)+(z,1)+(w,1)$.

Theorem [Denninghoff, Gatterdam 1989]
1. EH(mFIN,FIN)=RE.
2. H(mFIN,FIN) contains non-recursive languages.
3. There are regular languages which do not belong to H(mFIN,RE)

# Non-uniform splicing

$\gamma = (V, T, A, R)$

$\delta^0(A) = A$

$\delta^{k+1}(A) = \sigma(\delta^k(A), A)$

$$\delta^*(A) = \bigcup_{k\geq 0} \delta^k(A)$$

$L_n(\gamma) = \delta^*(A) \cap T^*$

Theorem [Mitrana, Petre, Rogojin 2010]

1. If $L \in$[E]$HG_n$(FIN,FIN), then $\$L\$ \in$[E]HG(FIN,FIN)
2. $EHG_n$(FIN,FIN) = EHG(FIN,FIN) = REG.

# Accepting splicing systems

$\gamma = (V, T, A, R, P)$

$\sigma^0(A,w) = A$, $\sigma^{k+1}(A,w) = \sigma^k(A,w) \cup \sigma(\sigma^k(A,w) \cup A)$

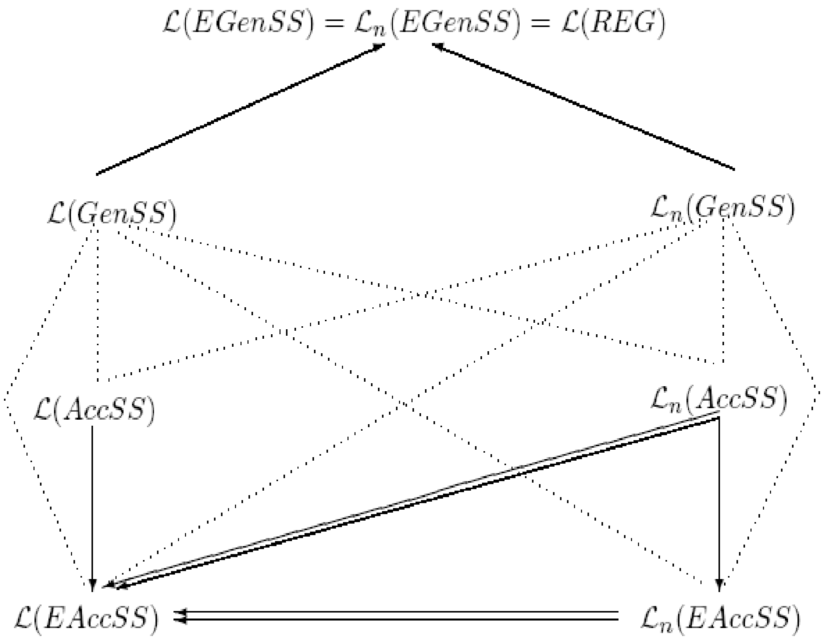
$$\sigma^*(A,w) = \bigcup_{k\geq 0} \sigma^k(A,w)$$

Uniform: w is accepted by $\gamma$ **iff** $\sigma^*(A,w) \cap P \neq \emptyset$

Non-uniform: w is accepted by $\gamma$ **iff** $\delta^*(A,w) \cap P \neq \emptyset$

# Accepting splicing systems (II)

Theorem [Mitrana, Petre, Rogojin 2010]



# Accepting splicing systems (III)

Decidability properties [Mitrana, Petre, Rogojin 2010]

1. The membership problem is decidable for the class EHA(FIN,FIN).

2. The membership problem is decidable for the class $HA_n$(FIN,FIN).

3. The finiteness problem is decidable iff the emptiness problem is decidable for $EHA_n$(FIN,FIN).

4. Both problems are decidable.

5. Let $L \in$ EHA(FIN,FIN). The problem "Is L finite?" is decidable iff the problem "Is card(L)$\leq$ k?" is decidable.

# Accepting splicing systems (IV)

Main disadvantage: Halting

# Accepting splicing systems (V)

## Enhanced variant:

$\gamma = (V, T, A, R, P)$

Uniform: w is accepted by $\gamma$ **iff** $\sigma^k(A,w) \cap P \neq \emptyset$
$\sigma^k(A,w) \cap F = \emptyset$

Non-uniform: w is accepted by $\gamma$ **iff** $\delta^k(A,w) \cap P \neq \emptyset$
$\delta^k(A,w) \cap F = \emptyset$

# Accepting splicing systems (VI)

Results:

1. $EHA_{[n]}(FIN,FIN) \subset fEHA_{[n]}(FIN,FIN)$.
2. For any regular language L, $\$L \in fEHA_n(FIN,FIN)$.
3. The class of regular languages is incomparable with fEHA(FIN,FIN).

A language L over V is called k-locally testable in the strict sense (k-LTSS for short) if there exists a triple $S_k = (A;B;C)$ over V such that for any w with $|w| \geq k$,

$$w \in L \text{ iff } [Pref_k(w) \in A;\ Suff_k(w) \in B;\ Inf_k(w) \in C]$$

L over V is preffix-disjoint if there exists a triple $S_k = (A;B;C)$ such that $L = L(S_k)$ and

$$(V^{-1}L) \cap (C \cup B) = \emptyset.$$

Theorem. Every preffix-disjoint or suffix-disjoint k-LTSS language belongs to $fEHA_n(FIN,FIN)$ for any $k \geq 1$.

# Accepting splicing systems (VII)

Decidability:

The membership problem is decidable for fEHA(FIN,FIN).

# Open problems

1. What is the computational power of $[f][E]HA_{[n]}$(FIN,FIN)? Is it more than that of a finite automaton?

2. Is the finiteness and/or emptiness problem decidable for EHA(FIN,FIN)?

3. Let $L \in$ EHA(FIN,FIN). The problem "Is L finite?" is decidable iff the problem "Is card(L)$\leq$ k?" is decidable. Which of them is decidable?

4. Which is the decidability status of the most important decision problems for $f[E]HA_{[n]}$(FIN,FIN)?

# Thank You