A Generalization of Haussler's Convolution Kernel - Mapping Kernel

Kilho (Yoshihiro) Shin

Carnegie Mellon CyLab Japan

Tetsuji Kuboyama

Gakushuin University

Haussler's Convolution Kernel

The original R-convolution kernel.

$$K(x,y)=\sum_{(x'_{1},\cdots,x'_{D},x)\in R}\ \sum_{(y'_{1},\cdots,y'_{D},y)\in R}\ \prod_{i=1}^{m} k(x'_{i},y'_{i})$$

χ： The space of data objects

$R\subseteq\underbrace{\chi'\times\cdots\times\chi'}_{D}\times\chi=(\chi')^{D}\times\chi$： A relation

$k:\chi'\times\chi'\to\mathbb{R}$： A symmetric function：

Haussler's Theorem

If $k(x', y')$ is positive semidefinite,

then, so is $K(x, y)$.

Reduction to the spacial case of D=1

$$K(x, y) = \sum_{(x', y') \in \chi_x \times \chi_y} k(x', y')$$

$$\chi_x = \{ x' | (x', x) \in R \}$$

Haussler's theorem for D > 1 is reduced to Haussler's theorem for D = 1.

The fundamental principle of convolution kernel

- Engineer new sophisticated kernels by tailoring simple and primitive kernels (*underlying kernel*) to complicated structures.
- Underlying kernels
  - e.g. Kronecker's delta function on labels
- Resulting kernels.
  - e.g. String kernels, Tree kernels

An example: the spectrum kernel

$$x = a_1 a_2 \cdots a_m \quad a_i \in \Sigma$$

$$y = b_1 b_2 \cdots b_n \quad b_j \in \Sigma$$

$$K(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} \delta(a_i, b_j)$$

Intuitively, positive semidefiniteness of $K(x, y)$ is derived from Haussler's theomrem, since every combination of characters is evaluated.

Formalization

$$x = \mathrm{AGTCTGA}, \qquad y = \mathrm{CGTATGC}$$

$$K(x, y) = 2\delta(\mathrm{A}, \mathrm{A}) + 4\delta(\mathrm{G}, \mathrm{G}) + 4\delta(\mathrm{T}, \mathrm{T}) + 2\delta(\mathrm{C}, \mathrm{C}) = 12$$

A single character pair is evaluated for multiple times. To apply Haussler's theorem, some formalization is necessary.

Formalization

$$\bar{\Sigma} = \Sigma \times \mathbb{N} \quad (\text{e.g.} \quad \Sigma = \{A, G, T, C\})$$

$$x = (A, 1)(G, 2)(T, 3)(C, 4)(T, 5)(G, 6)(A, 7)$$

$$\chi_x = \{(A, 1), (G, 2), (T, 3), (C, 4), (T, 5), (G, 6), (A, 7)\}$$

$$k : (\Sigma \times \mathbb{N}) \times (\Sigma \times \mathbb{N}) \rightarrow \mathbb{R} \text{ such that}$$

$$k((a, i), (b, j)) = \delta(a, b)$$

$$K(x, y) = \sum_{((a, i), (b, j)) \in \chi_x \times \chi_y} k((a, i), (b, j))$$

A variant

$$x = a_1 a_2 \cdots a_m \quad a_i \in \Sigma$$

$$y = b_1 b_2 \cdots b_n \quad b_j \in \Sigma$$

$$K_1(x, y) = \sum_{i=1}^{min\{m, n\}} \delta(a_i, b_i)$$

$K_1(x, y)$ proves to be positive semidefinite by Haussler's theorem by replacing $k((a,i),(b,j))$ with $\delta((a,i),(b,j))$.

Representation as a mapping kernel

Rather than replacing the underlying kernel, we modify the range of the summation.

$$K_1(x, y) = \sum_{\{((a,i),(b,j)) \mid i=j\} \subset \chi_x \times \chi_y} k((a,i),(b,j))$$

A question

$$K_1(x,y)=\sum_{\{((a,i),(b,j))|i=j\}\subset \chi_x\times\chi_y} k((a,i),(b,j))$$

is positive semidefinite.

The following are also positive semidefinite?

$$K_2(x,y)=\sum_{\{((a,i),(b,j)):i\equiv j \bmod 3\}\subset \chi_x\times\chi_y} k((a,i),(b,j))$$

$$K_3(x,y)=\sum_{\{((a,i),(b,j)):|i-j|=0 \text{ or } 3\}\subset \chi_x\times\chi_y} k((a,i),(b,j))$$

$$K_4(x,y)=\sum_{\{((a,i),(b,j)):|i-j|\leq 3\}\subset \chi_x\times\chi_y} k((a,i),(b,j))$$

Remark

The third kernel is also know as
the codon-improved kernel
[Zien et al., Bioinformatics, 2000].

$$K_3(x, y) = \sum_{\{((a,i),(b,j)) : |i-j| = 0 \text{ or } 3\} \subset \mathcal{X}_x \times \mathcal{X}_y} k((a,i),(b,j))$$

The question restated

$$K(x,y) = \sum_{(x',y') \in M_{xy} \subseteq \mathcal{X}_x \times \mathcal{X}_y} k(x',y')$$

Range	PSD
$M_{xy} = \mathcal{X}_x \times \mathcal{X}_y$	yes
$M_{xy} = \{((a,i),(b,j)) : i = j\}$	yes
$M_{xy} = \{((a,i),(b,j)) : i \equiv j \bmod 3\}$	?
$M_{xy} = \{((a,i),(b,j)) : \lvert i - j \rvert = 0 \text{ or } 3\}$	?
$M_{xy} = \{((a,i),(b,j)) : \lvert i - j \rvert \leq 3\}$	?

Definition - Transitivity

A family of subsets $\{M_{xy} \subseteq \chi_x \times \chi_y\}$ is transitive, if, and only if, the following conditions are met.

(1) $(x', y') \in M_{xy} \Rightarrow (y', x') \in M_{yx}$

(2) $(x', y') \in M_{xy} \wedge (y', z') \in M_{yz} \Rightarrow (x', z') \in M_{xz}$

Our main theorem
Less formal version

The following conditions are equivalent to each other.

(1) $\{M_{xy}\}$ is transitive.

(2) $K(x,y)=\sum_{(x',y')\in M_{xy}} k(x',y')$ is positive semidefinite for arbitrary underlying kernel $k(x',y')$.

Answer

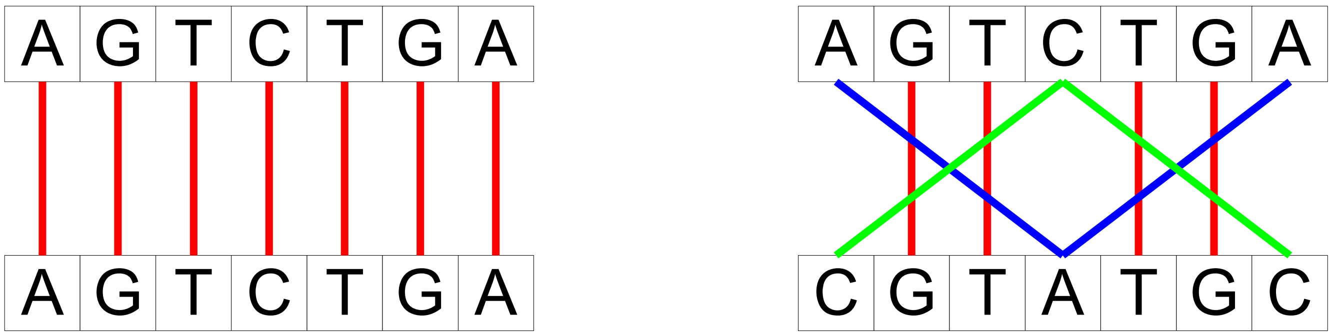
$$K(x, y) = \sum_{(x', y') \in M_{xy} \subseteq \chi_x \times \chi_y} k(x', y')$$

Range	PSD
$M_{xy} = \chi_x \times \chi_y$	yes
$M_{xy} = \{((a, i), (b, j)) : i = j\}$	yes
$M_{xy} = \{((a, i), (b, j)) : i \equiv j \bmod 3\}$	yes
$M_{xy} = \{((a, i), (b, j)) : \lvert i - j \rvert = 0 \text{ or } 3\}$	no for $\exists k$
$M_{xy} = \{((a, i), (b, j)) : \lvert i - j \rvert \leq 3\}$	no for $\exists k$

Example

$$x = \mathrm{AGTCTGA}, \qquad y = \mathrm{CGTATGC}$$

$$K_3(x,x) = K_3(y,y) = 7, \qquad K_3(x,y) = 8$$



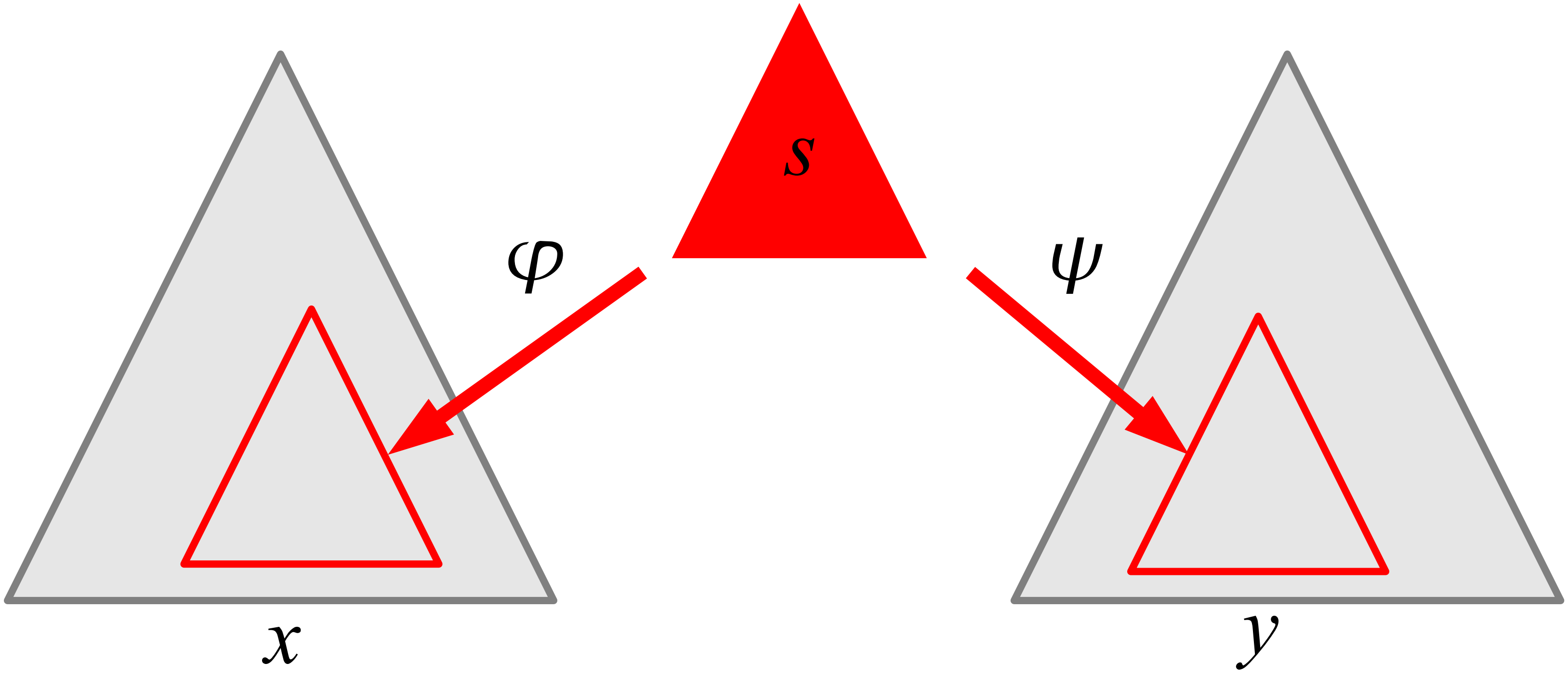
$$det\begin{pmatrix} K_3(x,x) & K_3(x,y) \\ K_3(y,x) & K_3(y,y) \end{pmatrix} = det\begin{pmatrix} 7 & 8 \\ 8 & 7 \end{pmatrix} = -15$$

Natural applications

- Size-of-index-structure-distribution (SISD) kernels
  - Distribution of the sizes of "index structures" that map onto substructures
- Edit-cost-distribution (ECD) kernels
  - Distribution of the costs of edit scripts

Index structures

φ, ψ: members of a predetermined class of embeddings (e.g. agreement subtrees)



(s, φ, ψ): An index structure

SISD kernels

χ_x: the set of all the substructures in x

C_{xy}: the set of all the index structures embedded in x and y

$f(X)$: a positive function

$\mathrm{size\_of}(s)$: the sizes of structures s

Then, the size-of-index-structure-distribution kernel is defined as follows.

$$K(x, y) = \sum_{s \in C_{xy}} f(\mathrm{size\_of}(s))$$

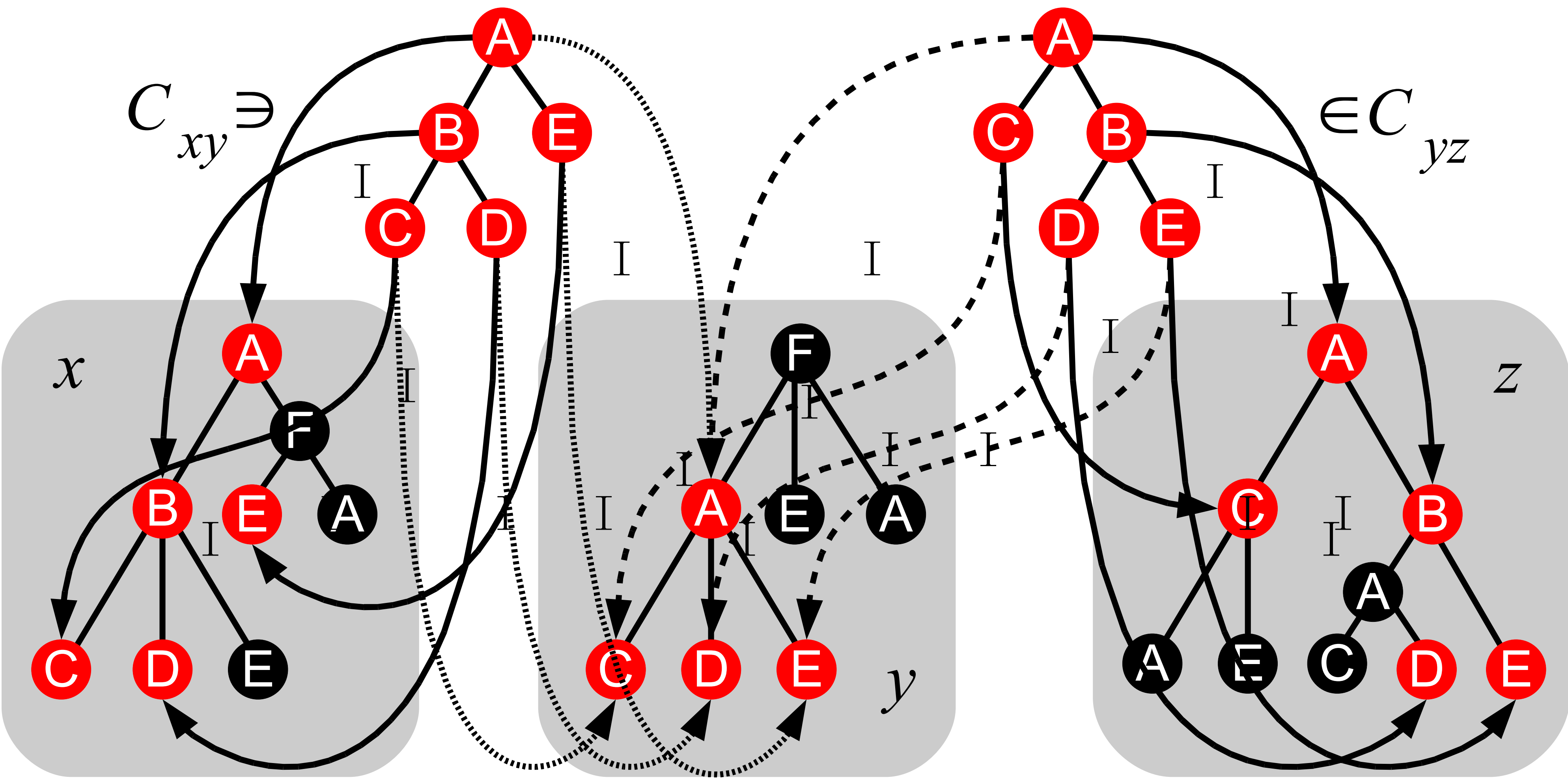
SISD kernels

The function $f(\mathsf{size\_of}(s))$ is always positive semidefinite.

In contrast, C_{xy} is not necessarily transitive.
e.g. Compatible subtree (see next slide)

Compatible subtrees

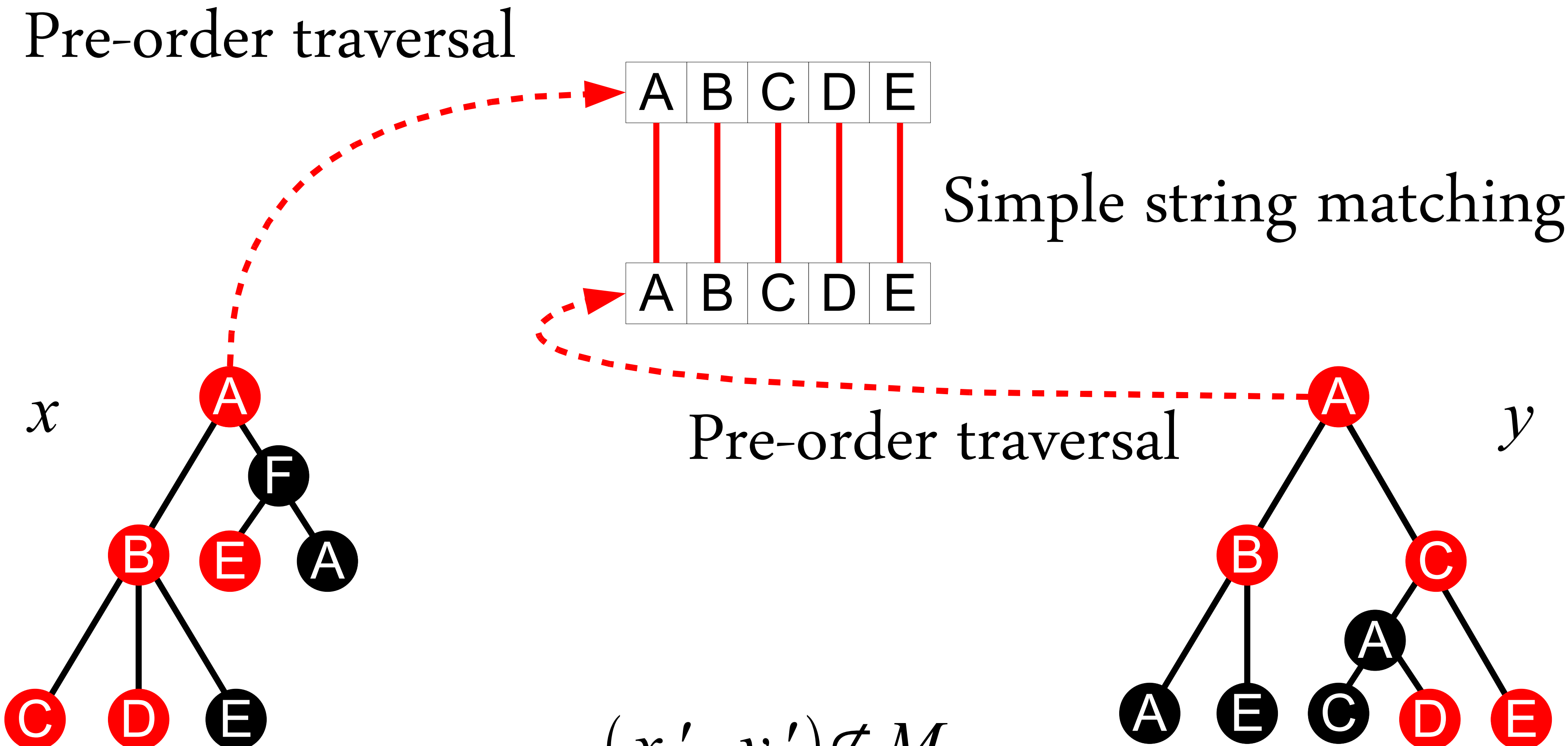
Contraction of edges is allowed.



An advantage of the mapping kernel

- Even for more complicated structures, we can take advantage of simple underlying kernels instead of engineering underlying kernels taking the structures into account.
- Constraints caused from structures can be included in the definition of *mappings.*

An example



$$(x', y') \notin M_{xy}$$

Conclusions

- We have generalized Haussler's convolution kernel, and have proposed the mapping kernel.
- Our main theorem gives a necessary and sufficient condition that mapping kernels become positive semidefinite for arbitrary underlying kernels.
- Haussler's theorem is a special case of our main theorem.

Polynomial kernels

If a polynomial

$$F(X)=a_n X^n + a_{n-1} X^{n-1} + \cdots + a_1 X + a_0$$

has only non-negative coefficients,

$$K(x, y)=F(k(x, y))$$

proves to be positive semidefinite for arbitrary positive semidefinite underlying kernels $k(x, y)$.

Our generalization (reported at ALT 2007)

- Generalization in two direction
  - To allow multiple variables
  - To allow negative coefficients
- If the "coefficient matrix" of a polynomial is positive semidefinite, then the resulting "polynomial kernel" is also positive semidefinite for arbitrary positive semidefinite underlying kernels.