CZYSZCZENIE DANYCH:

Automatyczny podział tekstu na rekordy o określonej strukturze

(na podstawie pracy: „*Automatic segmentation of text into structured records*”, Borkar, Deshmukh, Sarawagi, SIGMOD 2001)

Wojtek Kulik

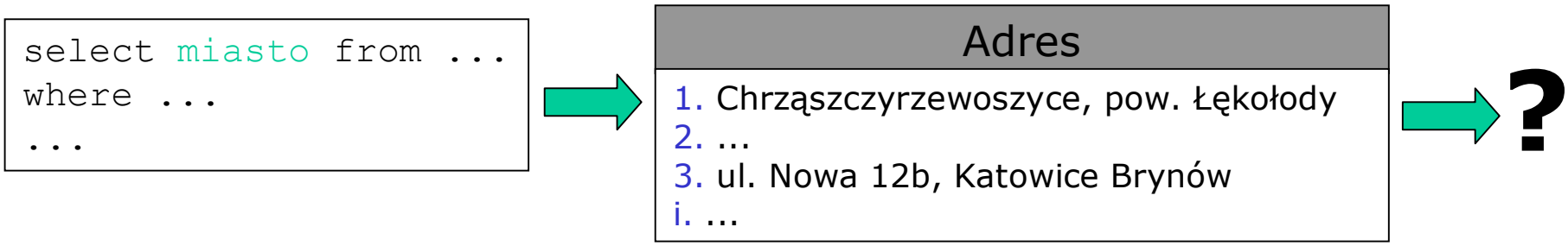
17 marca 2005

Wstęp

- Czyszczenie danych
- PROBLEM: Podział ciągłego tekstu na rekordy o określonej strukturze
- Ukryte łańcuchy Markova jako przykład rozwiązania problemu:
  - Podstawowy model
  - Rozszerzenia
- Pomysł na pracę magisterską

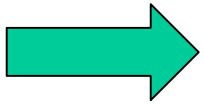
Po co czyścić dane? (1)

- Mamy gdzie przechowywać dane: **bazy danych**
- Mamy czym analizować dane: **języki zapytań**
- PROBLEM: dane ***nie bardzo*** dają się przepytywać w formie, w jakiej znajdują się w bazie danych



Po co czyścić dane? (2)

```
select osoba from ...
where kwota =
(select max(kwota) ...)
...
```



Osoba	Kwota
Jan Kowalski	10
Kowalski J.	300
Pan Jan Adam Kowalski	70

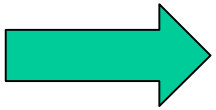
- Barbra Streisand ☺
- Ubezpieczenia samochodów – (bez gwarancji...)

Czyszczenie danych (1)

- Najczęściej, proces czyszczenia danych rozumie się jako cały proces przygotowania danych do analizy zakończony eliminacją duplikatów (danych logicznie tożsamych)

Osoba	Kwota
Jan Kowalski	10
Kowalski J.	300
Pan Jan Adam Kowalski	70

czyszczenie



Osoba	Kwota
Pan Jan Kowalski	380

Efekt:
- dokładniejsza analiza
- szybsze przetwarzanie (mniej rekordów)

Czyszczenie danych (2)

Czyszczenie wykonuje się w trzech etapach:

1. Podział ciągłego tekstu na rekordy o określonej strukturze (ang. *elementization*, „parsowanie”)
2. Standaryzacja (m. in.: ujednolicanie form, poprawa literówek)
3. Identyfikacja i eliminacja duplikatów (ang. *matching*, *householding*)

Czyszczenie danych (3)

1. „Parsowanie”

Osoba	Kwota
Jan Kowalski	10
Kowalski J.	300
Pan Jan Adam Kowalski	70

Przedrostek	Imię	Drugie Imię	Nazwisko	Kwota
	Jan		Kowalski	10
	J.		Kowalski	300
Pan	Jan	Adam	Kowalski	70

Czyszczenie danych (4)

2. Standaryzacja

Przedrostek	Imię	Drugie Imię	Nazwisko	Kwota
	Jan		Kowalski	10
	J.		Kowalski	300
Pan	Jan	Adam	Kowalski	70

Przedrostek	Imię	Drugie Imię	Nazwisko	Kwota
Pan	Jan		Kowalski	10
Pan	J.		Kowalski	300
Pan	Jan	Adam	Kowalski	70

Czyszczenie danych (5)

3. Eliminacja duplikatów

Przedrostek	Imię	Drugie Imię	Nazwisko	Kwota
Pan	Jan		Kowalski	10
Pan	J.		Kowalski	300
Pan	Jan	Adam	Kowalski	70

Osoba	Kwota
Pan Jan Kowalski	380

„Parsowanie” (1)

- Dane przechowywane jako ciągły tekst, np. adresy, dane osobowe, dane bibliograficzne
- Wygodniej jest je wprowadzać w postaci ciągłego tekstu
- Dane w jednej tabeli mogą być wprowadzane przez wiele osób, pochodzić z różnych tabel, formularzy internetowych, itp.

- Analiza danych wymaga specjalnej postaci danych – porównywanie całych ciągów znaków nie ma sensu

„Parsowanie” (2)

- Takie dane mają niejawną strukturę, mimo iż brak w nich jawnego separatora pomiędzy polami rekordów
- Poszczególne rekordy mogą się różnić np. kolejnością występowania pól
- Nie wszystkie pola muszą być obecne w każdym rekordzie
- Nasza wiedza sprawia, że jesteśmy w stanie wychwycić ich strukturę „automagicznie”
- Istniejące narzędzia bazują na pewnej liczbie zaszytych na sztywno reguł

„Parsowanie” (3)

- To nie jest przetwarzanie języka naturalnego! (złożona struktura)
- To nie jest ekstrakcja danych z HTML! (bardzo regularna struktura)

Przykłady:

1) Adresy

„Parsowanie” (4)

2) Bibliografia (CiteSeer)

Patricia G. Selinger, et al.. Access path selection in a relational database management system. In Proceedings of the ACM SIGMOD Conference, pages 23-34, 1979.

McGraw-Hill. Selinger P.; Astrahan, M.; Chamberlin, D.; Lorie, R.; and Price, T. 1979. Access path selection in a relational database management system. In SIGMOD ’79.

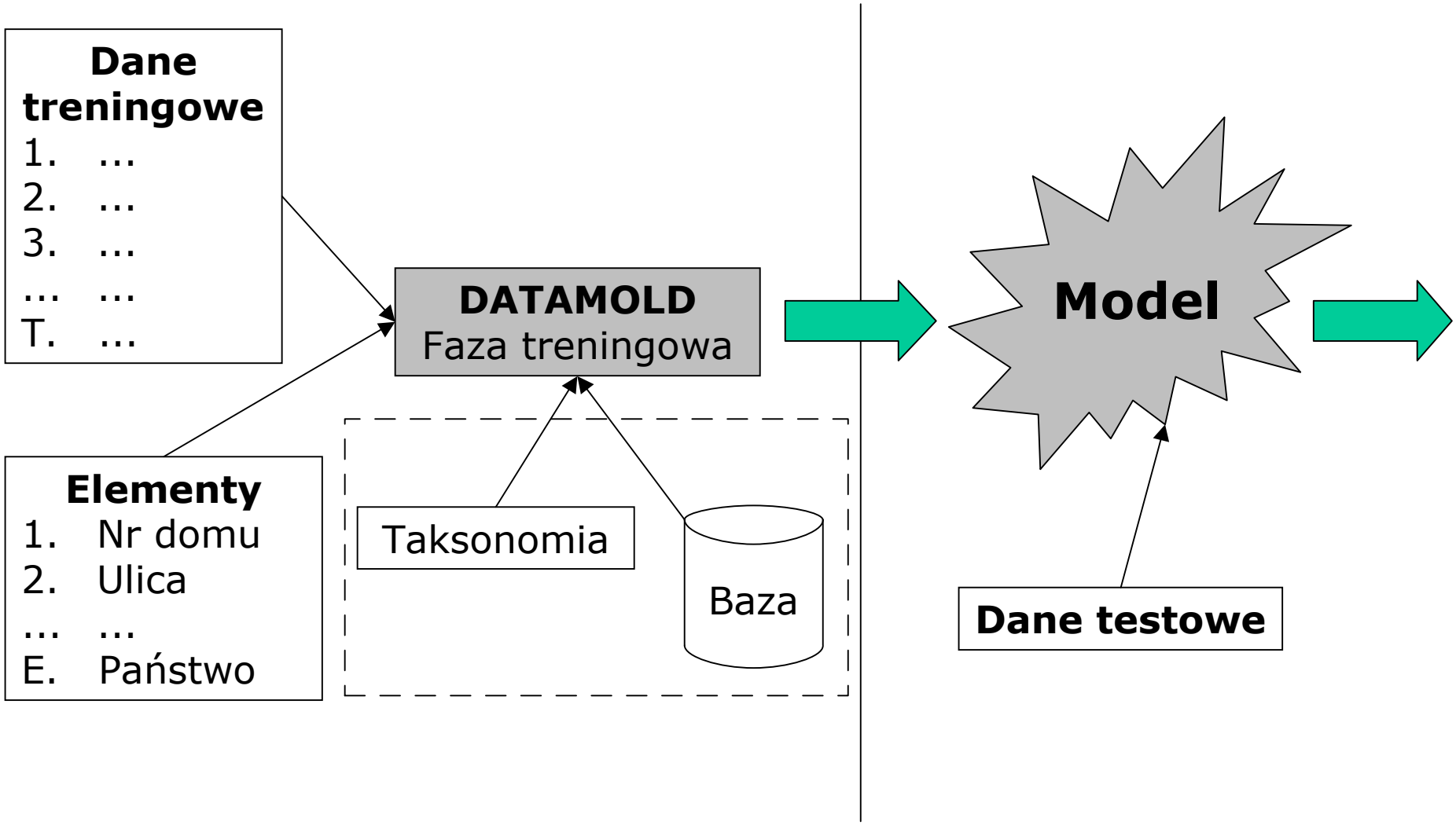
Praca o optymalizacji zapytań (Selinger, 1979) pojawia się 283 razy jako 13 różnych prac.

Rozwiązanie problemu

- Rozszerzony ukryty łańcuch Markova (ang. *Hidden Markov Model*, HMM)
- HMM - probabilistyczny automat skończony

- DATAMOLD – zastosowanie HMM do rozbijania adresów na części składowe (ang. *address elementization*)
  - Dane treningowe
  - Dane testowe

DATAMOLD – architektura



HMM do „parsowania” (1)

- *n* stanów + 2 wyróżnione (START i KONIEC)
- *m* symboli wyjściowych
- macierz *A* przejść pomiędzy stanami

  a_{ij} = prawdopodobieństwo przejścia ze stanu i do stanu j
- macierz *B* akceptacji (emisji)

  b_{jk} = prawdopodobieństwo akceptacji (emisji) k-tego symbolu wejściowego w stanie j
- 1 stan odpowiada jednemu elementowi
- element odpowiada 1 polu rekordu
- 1 element może odpowiadać wielu tokenom wejścia

HMM do „parsowania” (2)

- ścieżka – ciąg przejść pomiędzy stanami + akceptacja symbolu w każdym stanie
- HMM akceptuje ciąg symboli $o_1 o_2 ... o_k$ jeśli istnieje ścieżka (k+1-stanowa) od stanu START do stanu KONIEC, która zostanie wybrana z prawdopodobieństwem >0
- W ogólności może istnieć więcej niż jedna taka ścieżka
- Interesuje nas NAJBARDZIEJ PRAWDOPODOBNA ścieżka

HMM do „parsowania” (3)

- Akceptacja HMM:

WEJŚCIE: $o_1 o_2 ... o_k$ (ciąg symboli)

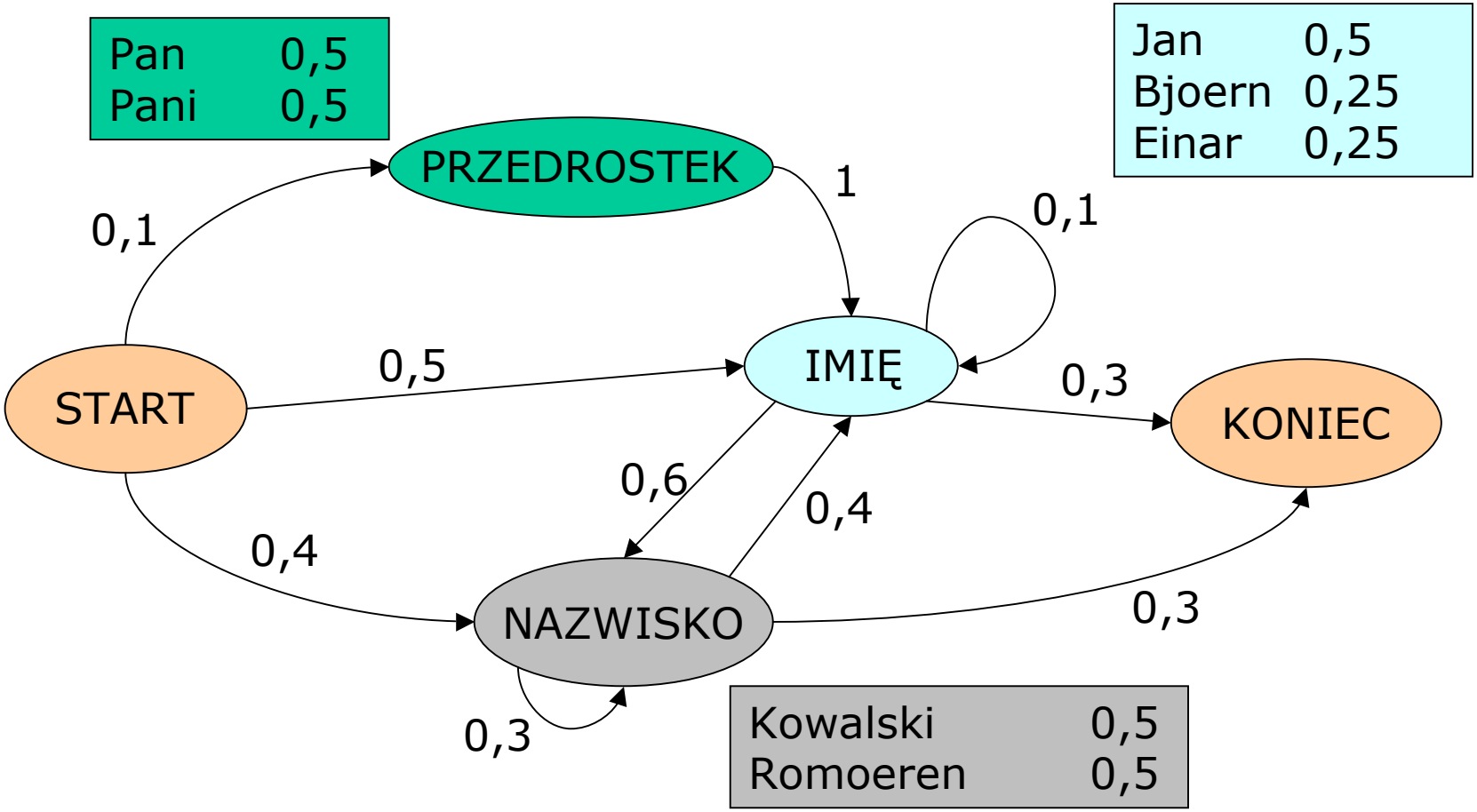
UWAGA: musimy wiedzieć, jak oddzielane są symbole!

WYJŚCIE: $<o_1, E_{i1}>, <o_2, E_{i2}>, ..., <o_k, E_{ik}>$

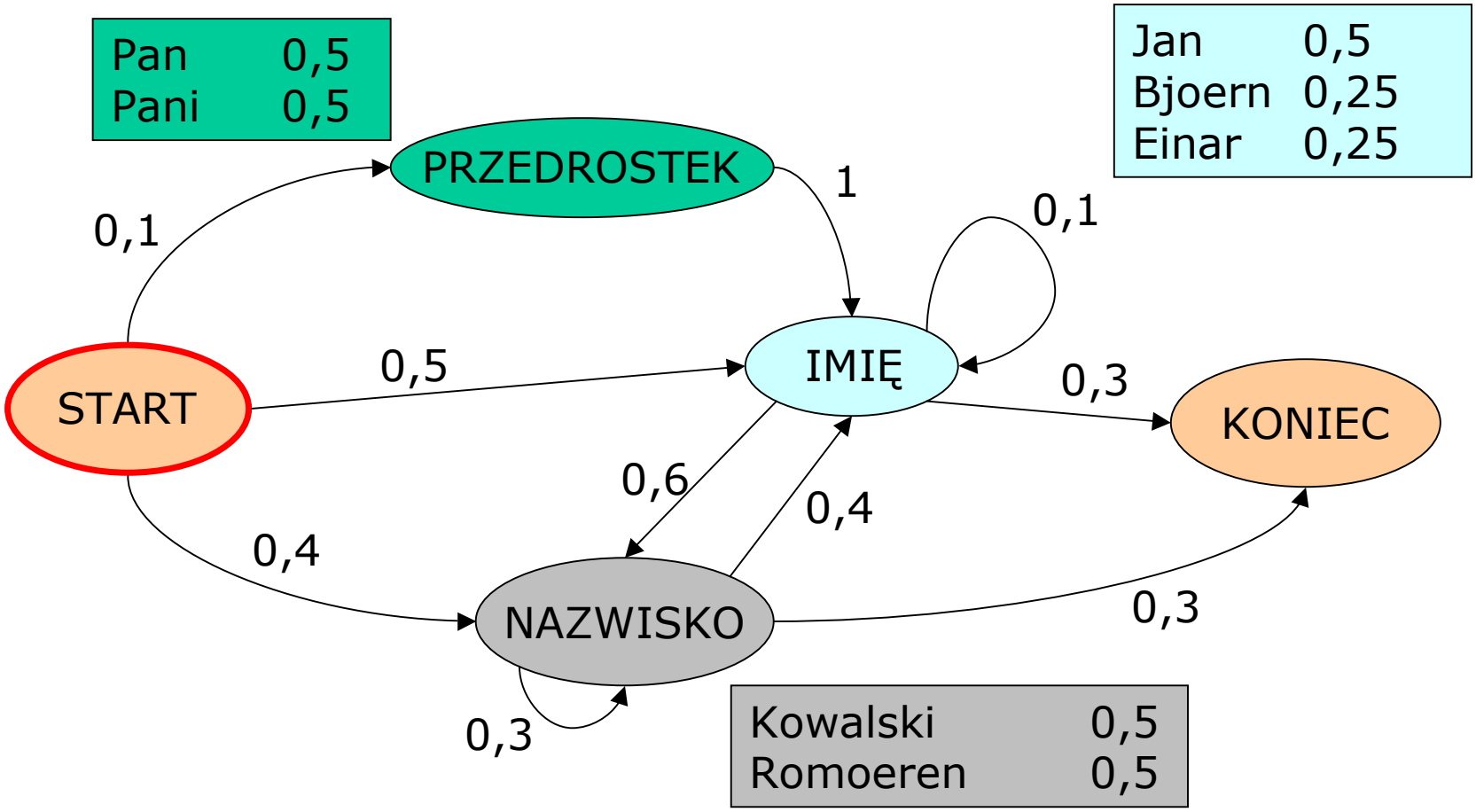
(ciąg par: <*symbol*, *element*>)

(lub OOOOPS...)

Przykładowy model (1)

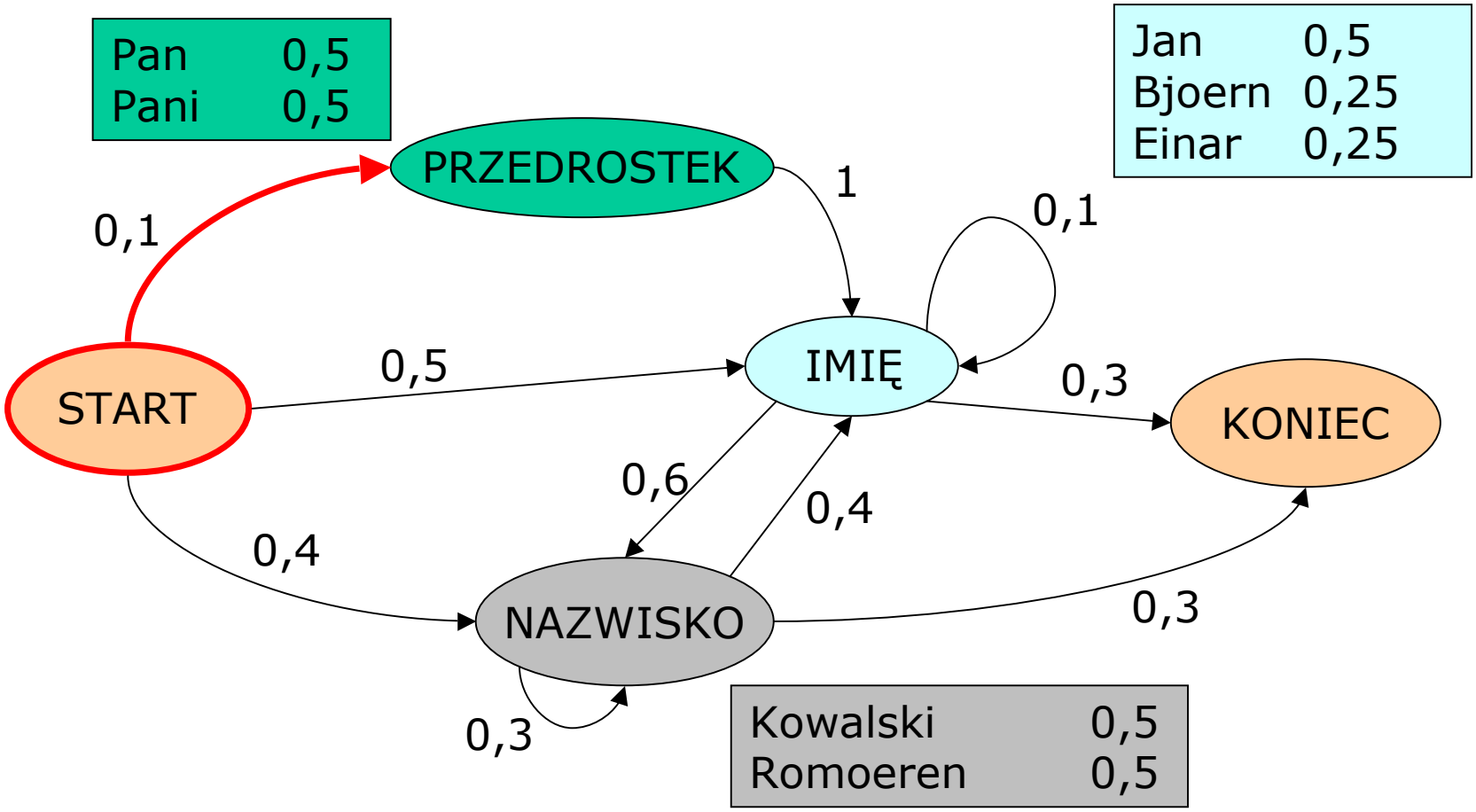


Przykładowy model (2)

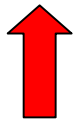


Wejście: JAN#KOWALSKI Wyjście:

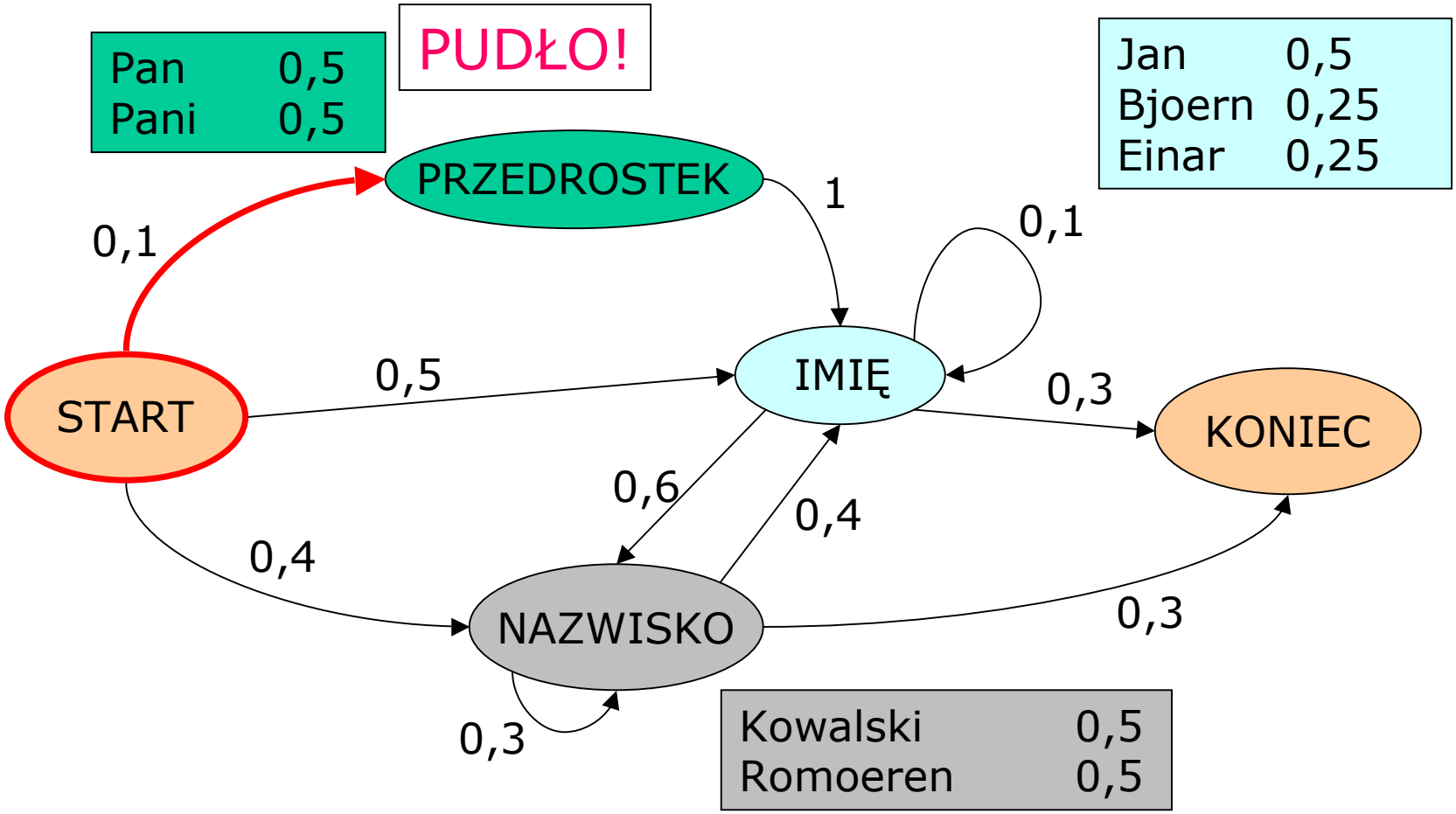
Przykładowy model (2)



Wejście: JAN#KOWALSKI Wyjście:

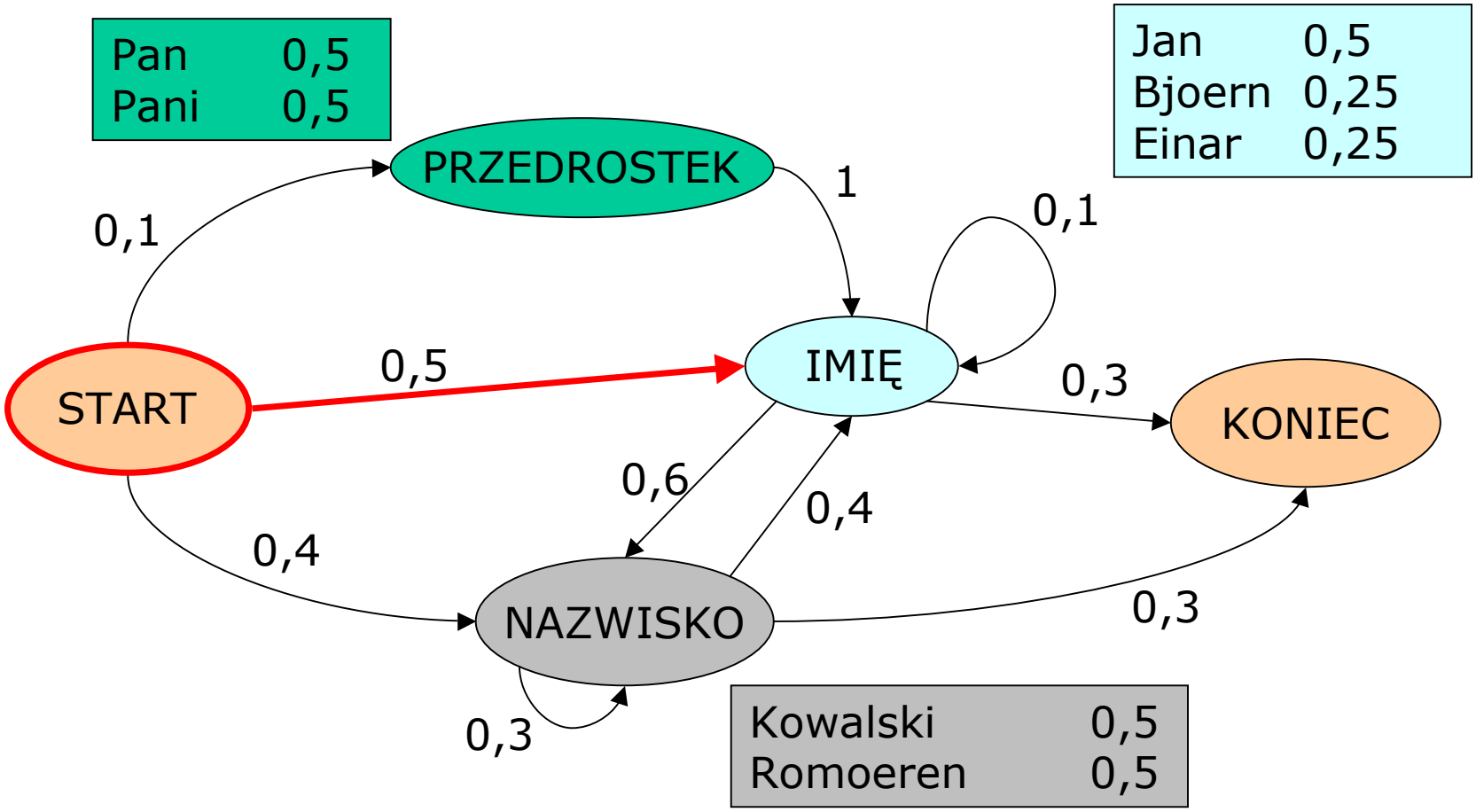


Przykładowy model (2)



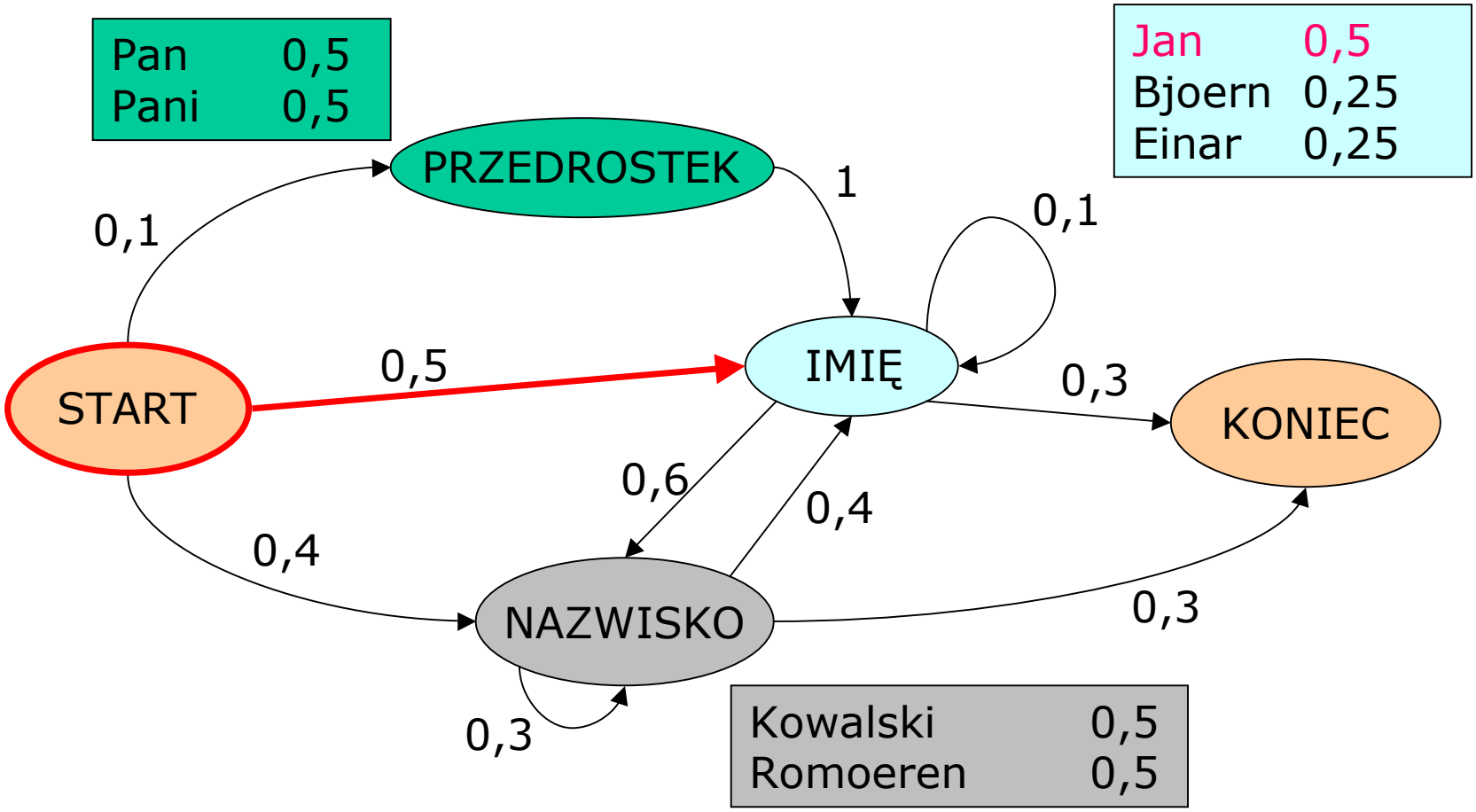
Wejście: JAN#KOWALSKI Wyjście:

Przykładowy model (2)



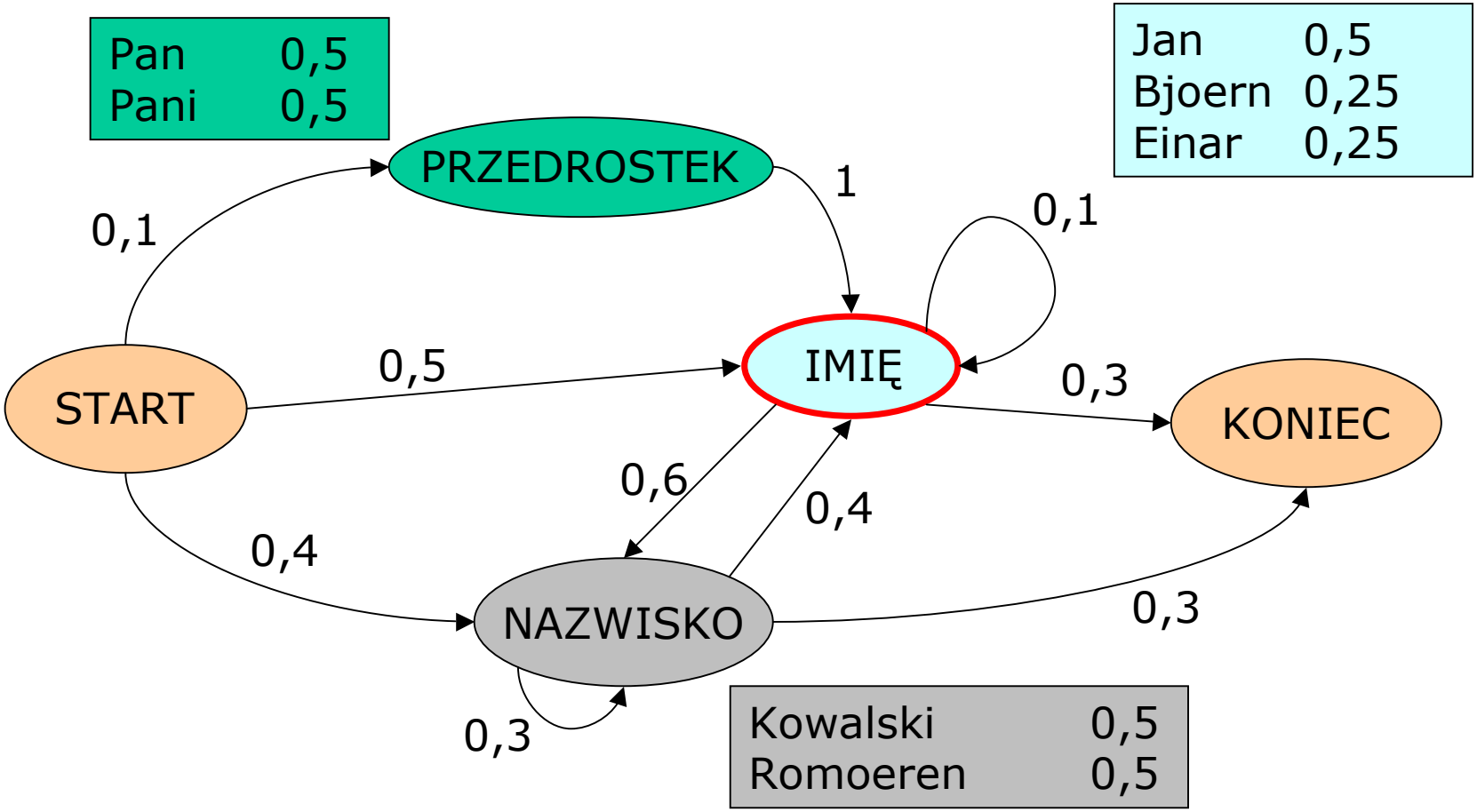
Wejście: JAN#KOWALSKI Wyjście:

Przykładowy model (2)



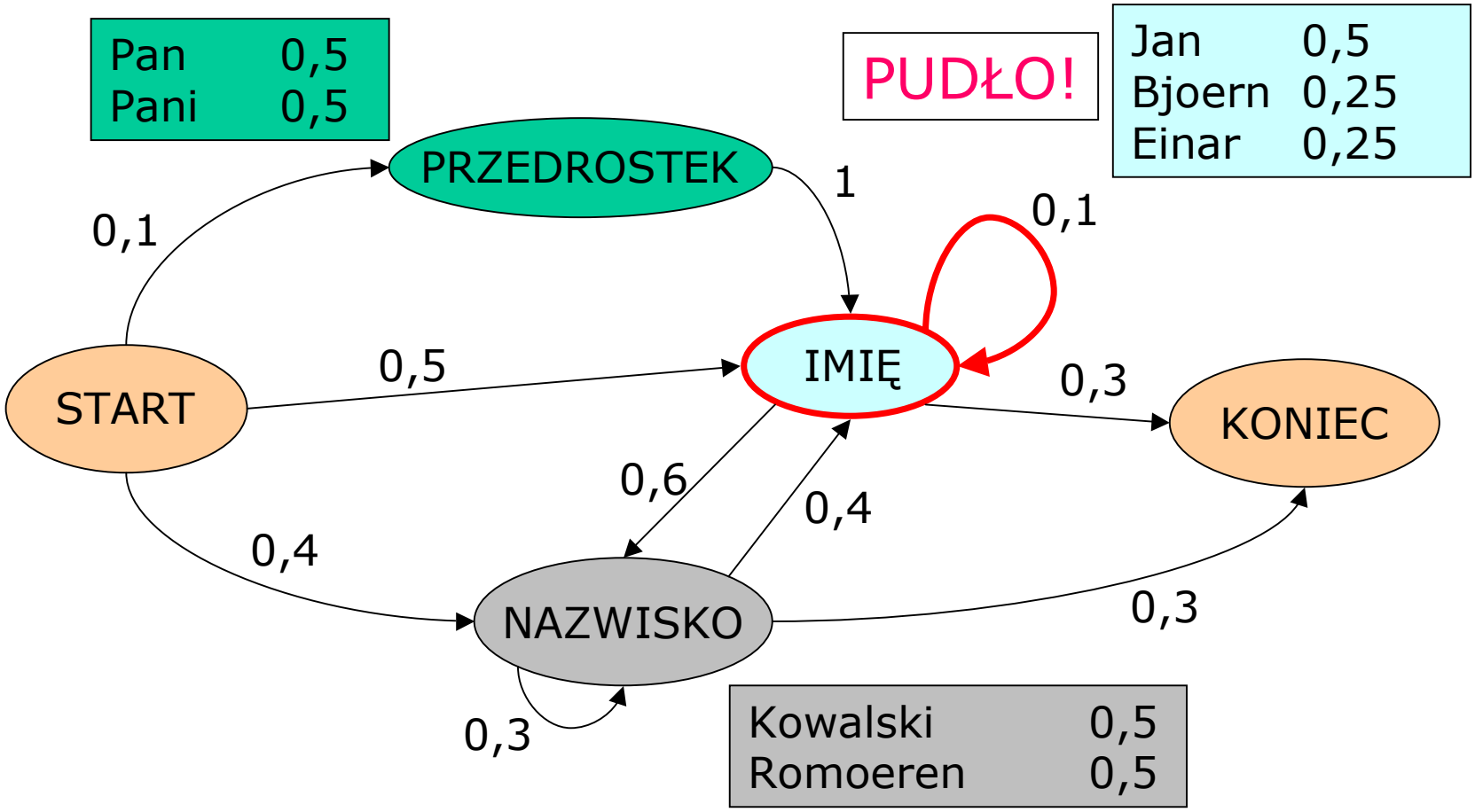
Wejście: JAN#KOWALSKI Wyjście:

Przykładowy model (2)



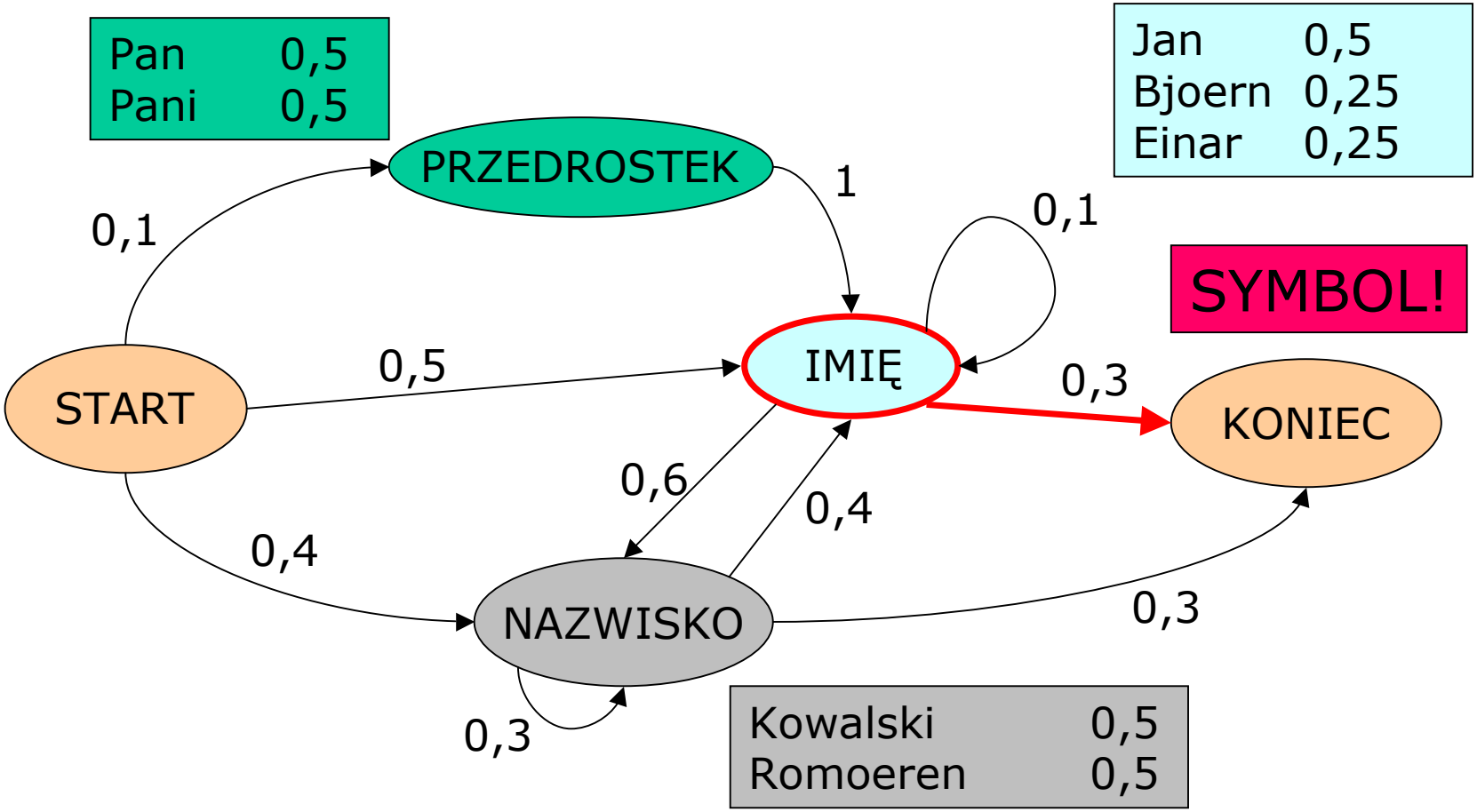
Wejście: JAN#KOWALSKI Wyjście: <„JAN”, IMIĘ>

Przykładowy model (2)



Wejście: JAN#KOWALSKI Wyjście: <„JAN”, IMIĘ>

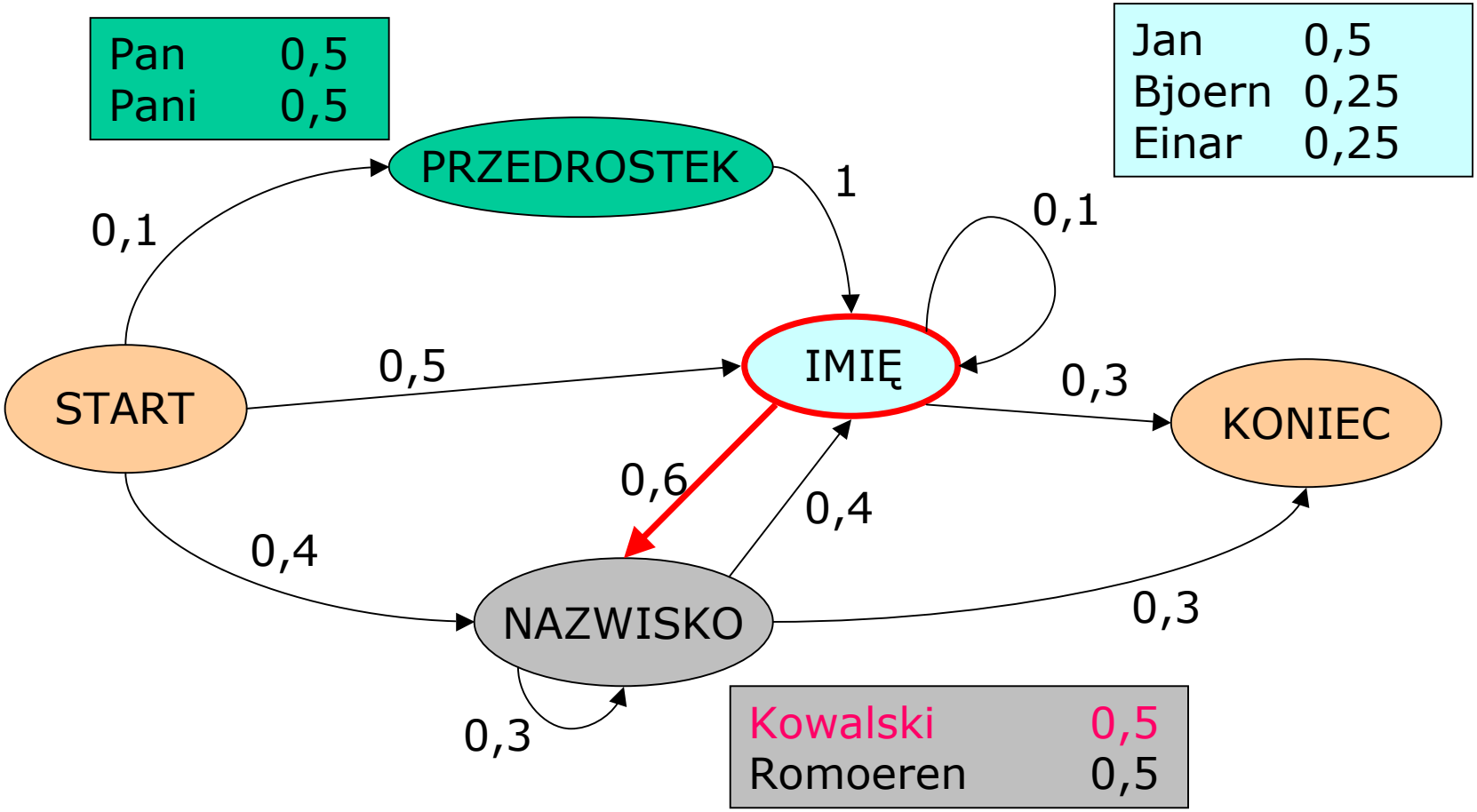
Przykładowy model (2)



Wejście: JAN#KOWALSKI

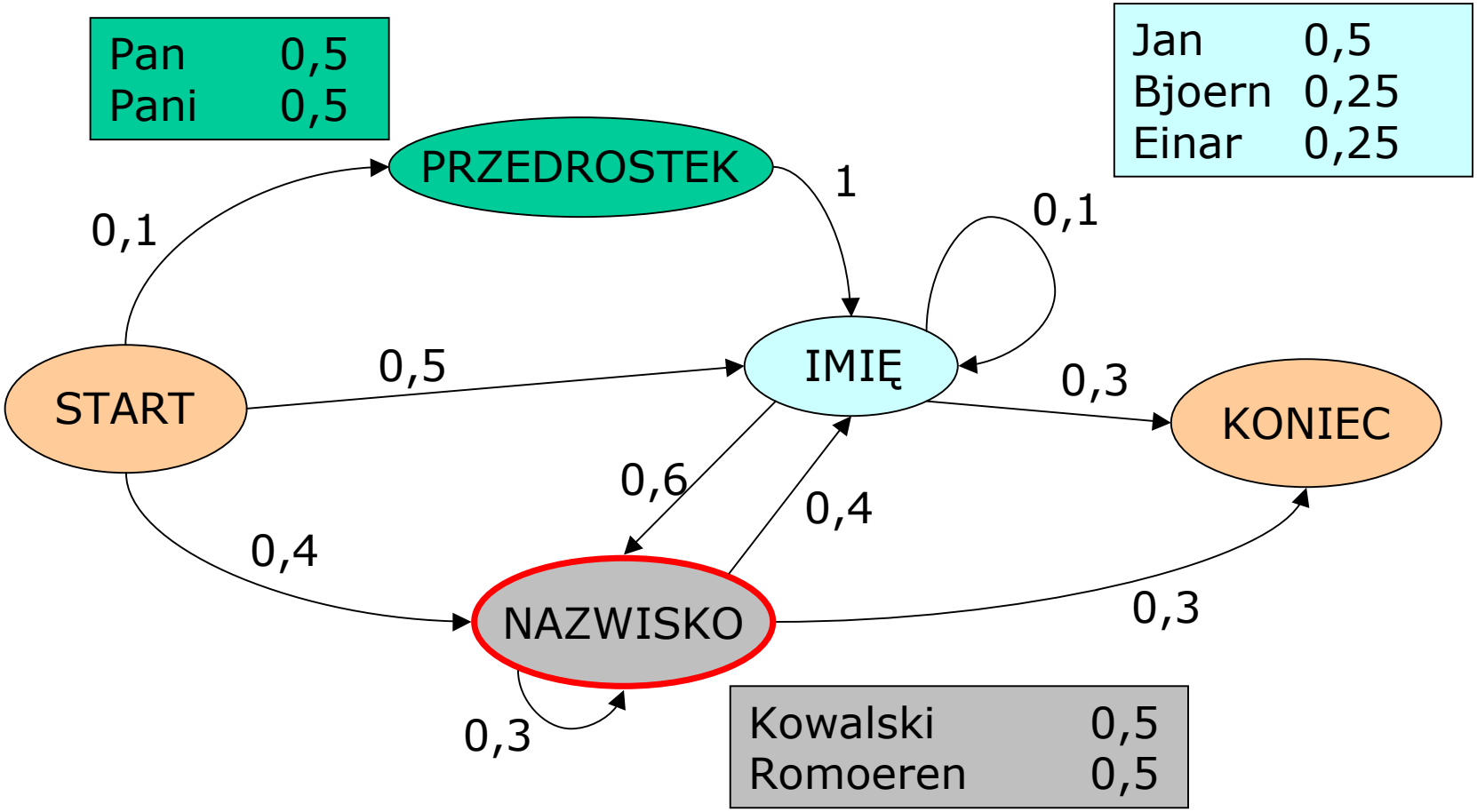
Wyjście: <„JAN”, IMIĘ>

Przykładowy model (2)



Wejście: JAN#KOWALSKI Wyjście: <„JAN”, IMIĘ>

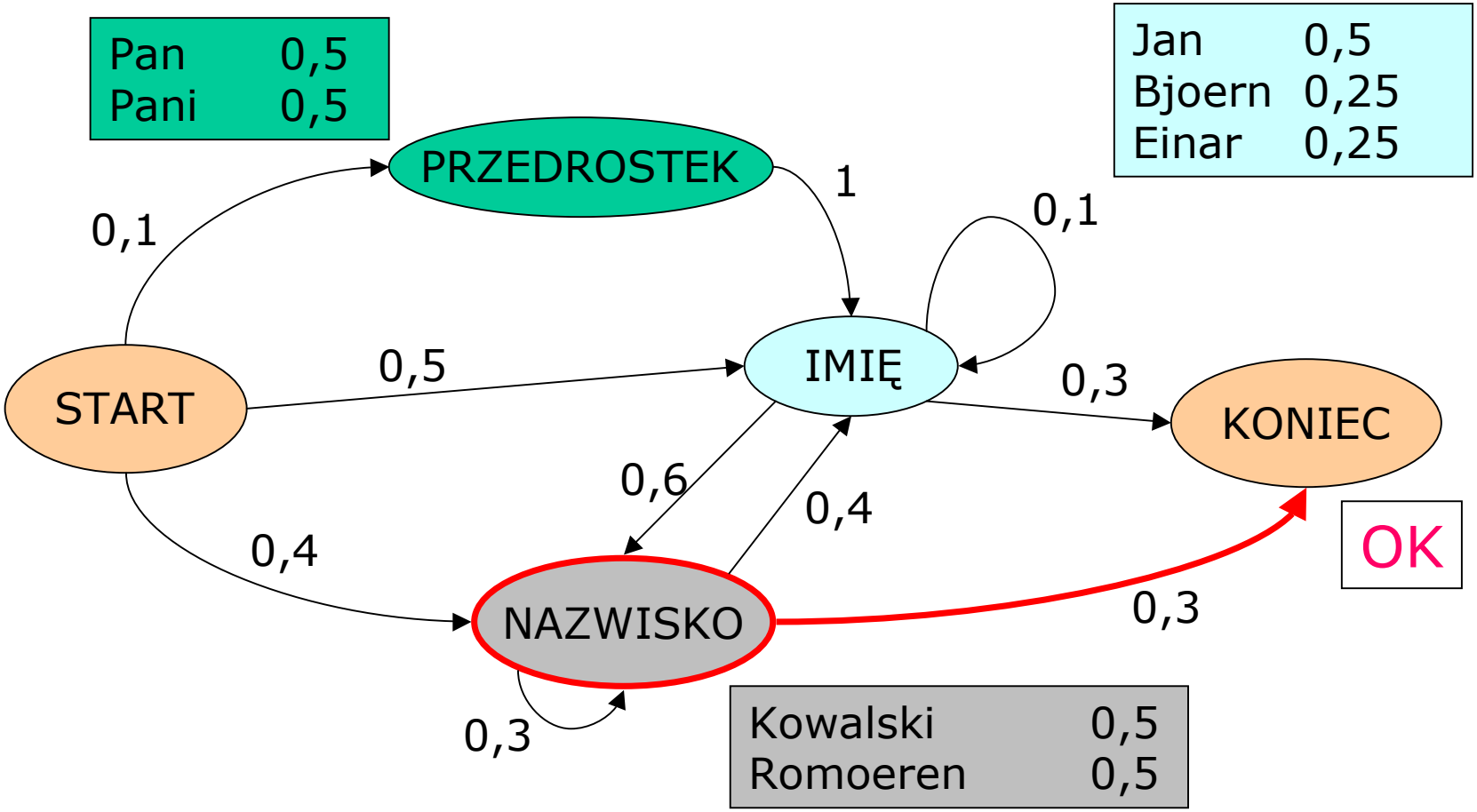
Przykładowy model (2)



Wejście: JAN#KOWALSKI

Wyjście: <„JAN”, IMIĘ>, <„KOWALSKI”, NAZWISKO>

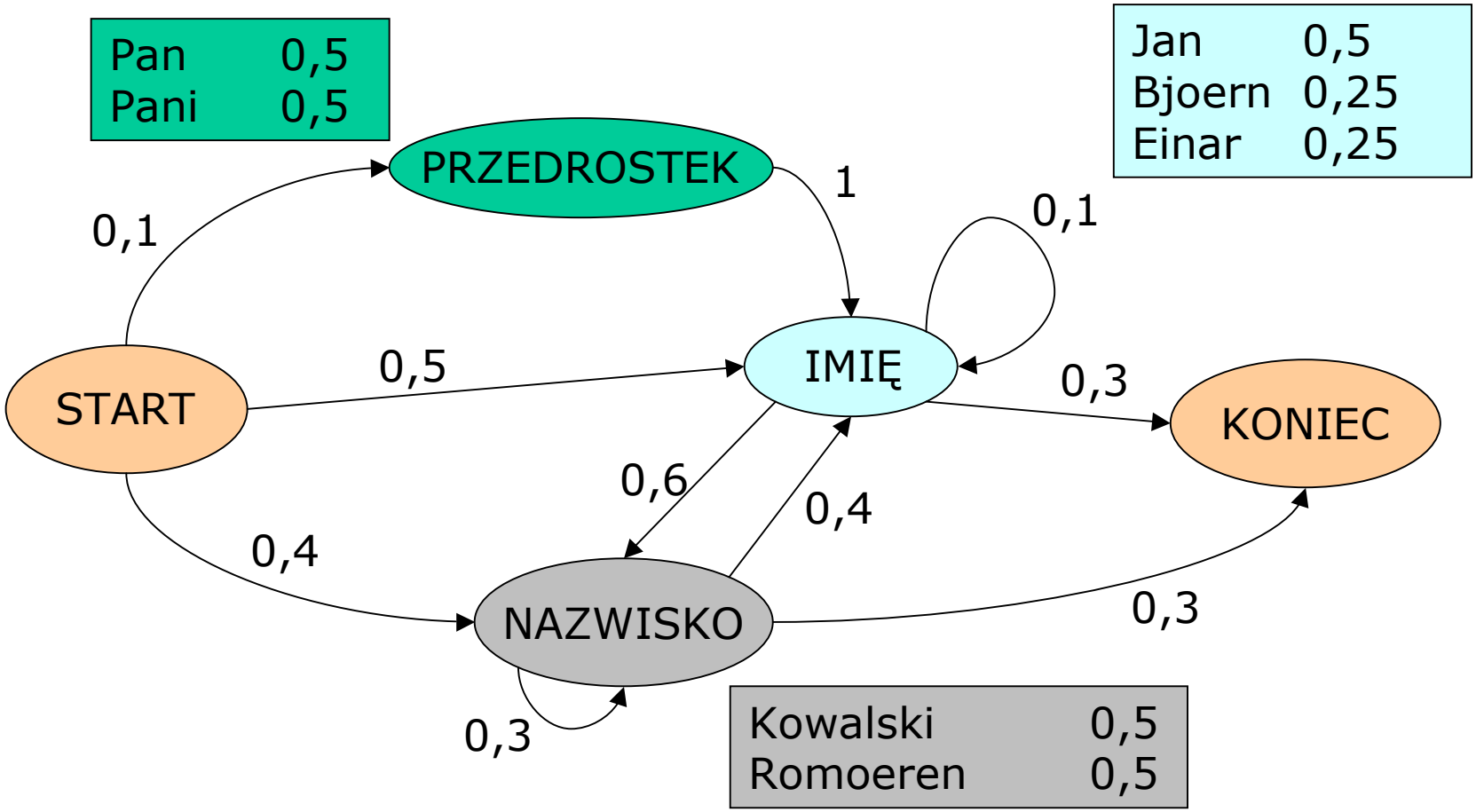
Przykładowy model (2)



Wejście: JAN#KOWALSKI

Wyjście: <„JAN”, IMIĘ>, <„KOWALSKI”, NAZWISKO>

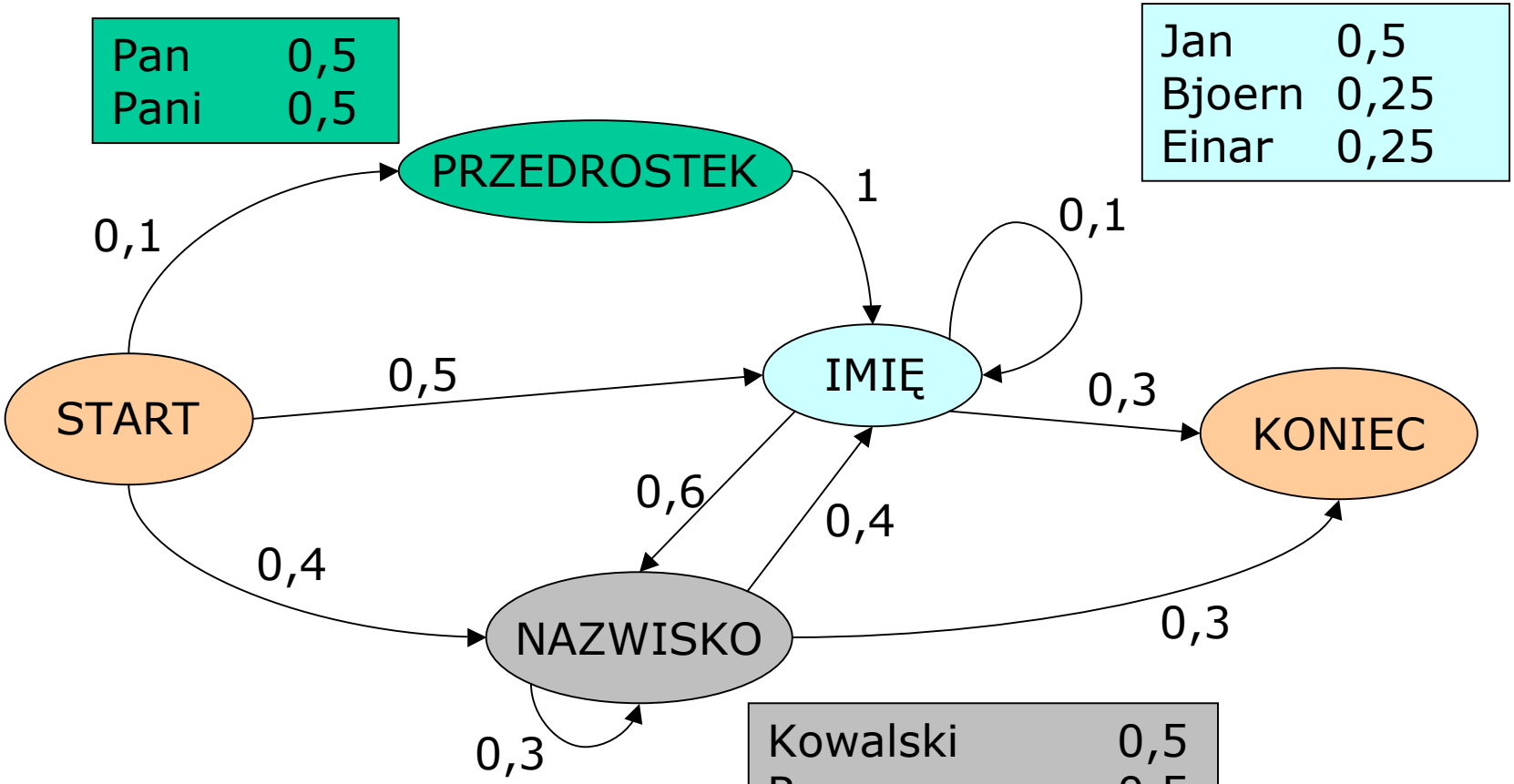
Przykładowy model (2)



Wejście: JAN#KOWALSKI

Wyjście: <„JAN”, IMIĘ>, <„KOWALSKI”, NAZWISKO>

Przykładowy model (2)

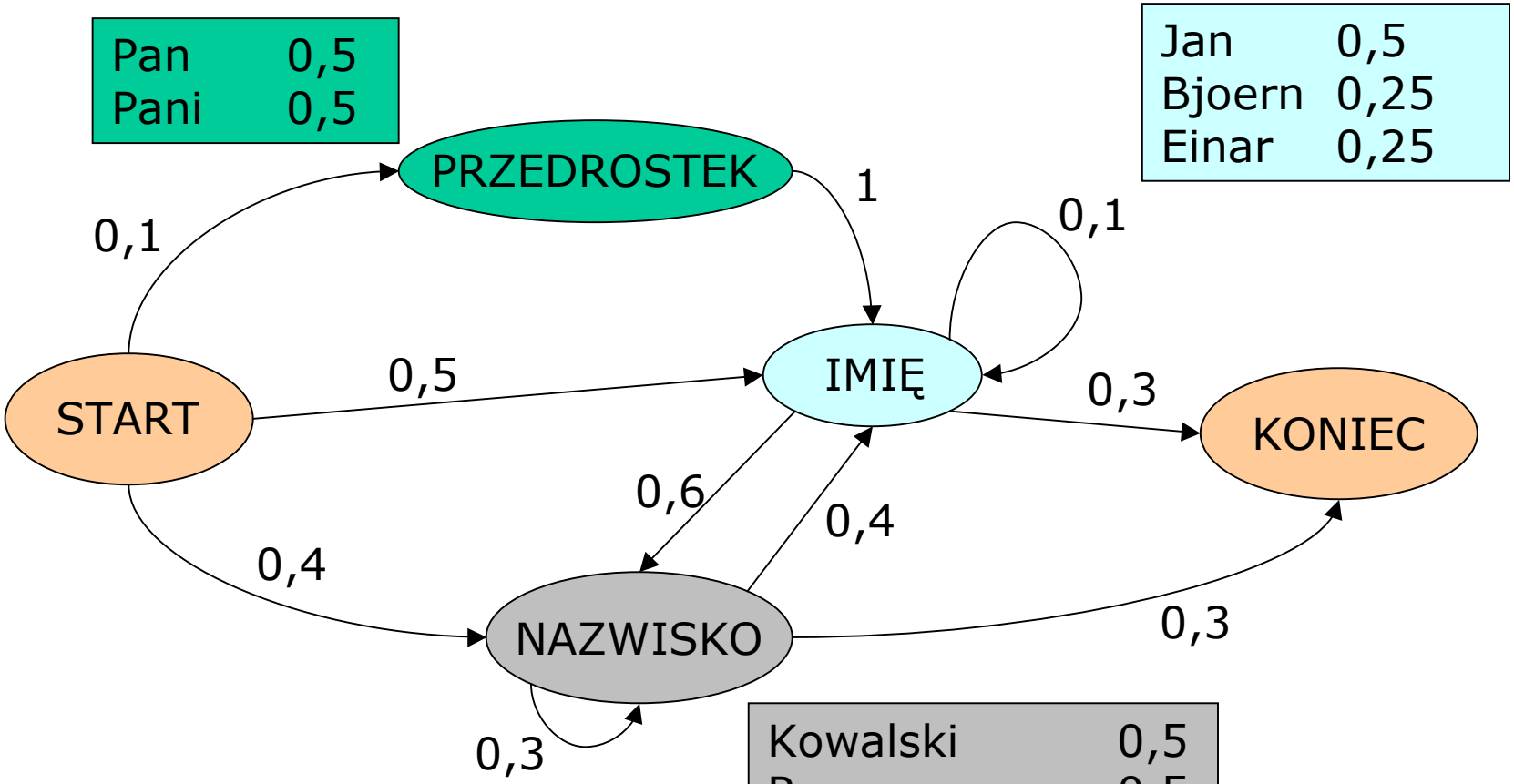


PODOBNIE:

Wejście: KOWALSKI#JAN

Wyjście: <„KOWALSKI”, NAZWISKO>, <„JAN”, IMIĘ>

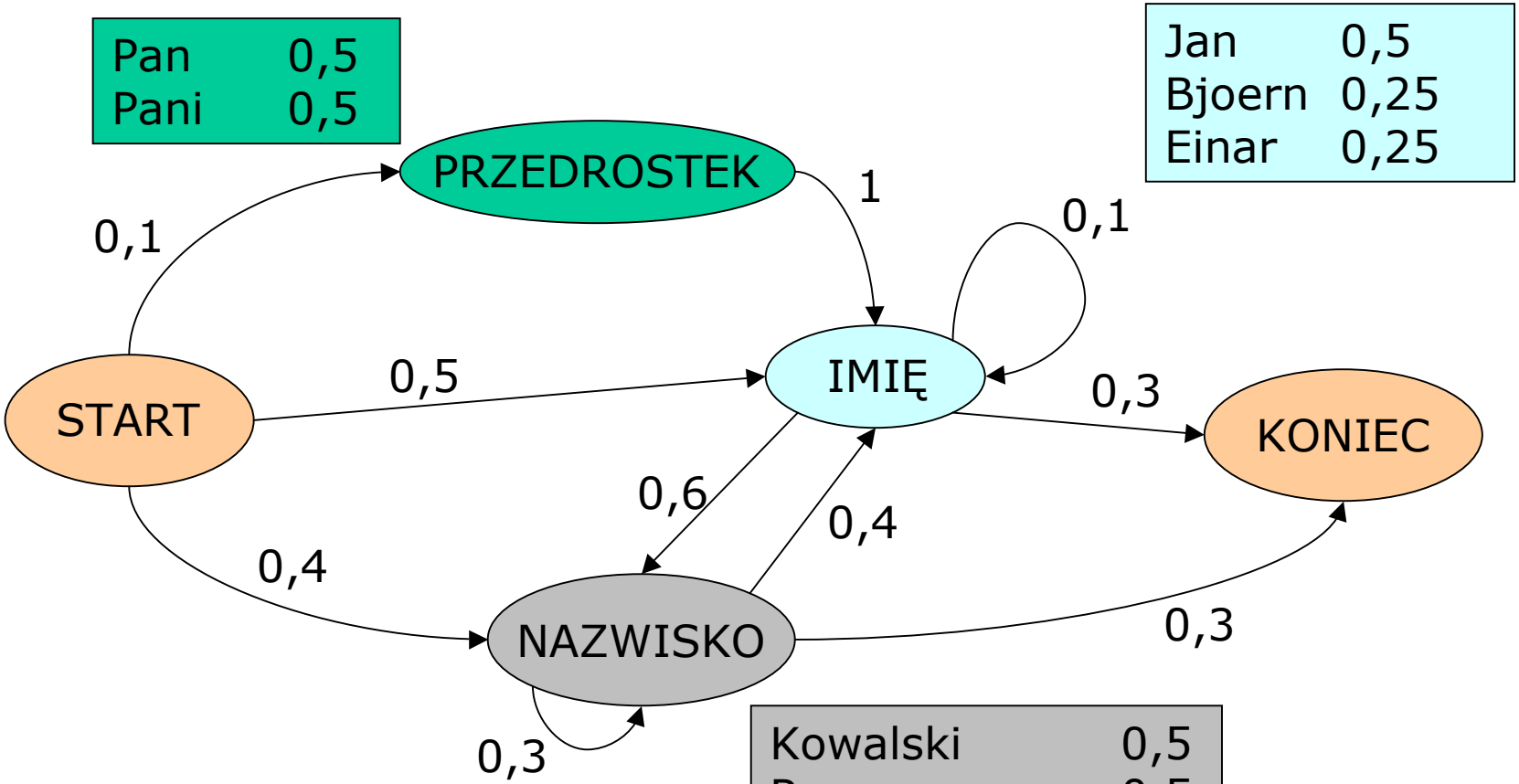
Przykładowy model (2)



PODOBNIE:

Wejście: PAN#JAN#KOWALSKI Wyjście: ...
Wejście: Bjoern#Einar#Roemoren Wyjście: ...

Przykładowy model (2)



NIESTETY:

Wejście: PAN#ADAM#NOWAK
Wejście: Ronaldo#Luis#Nazario#de#Lima

OOOOPS

Przykładowy model – pytania

- Jak trenować model?
- Jak szukać rozwiązania?
- Co gdy zawiodą słowniki?
- Co z kolejnością wewnątrz jednego elementu?

Jak trenować model? (1)

- Trening przebiega dwuetapowo:
  1. Określenie struktury HMM (liczba stanów oraz istotne krawędzie łączące stany) oraz utworzenie słowników.

     Liczbę stanów (tj. elementów) musimy znać z góry.

     Słowniki tworzy się na podstawie danych treningowych.

     Dane treningowe to zestaw par <*symbol*, *element*> utworzonych ręcznie ☹.

Jak trenować model? (2)

- Trening przebiega dwuetapowo:
  2. Określenie prawdopodobieństw przejść pomiędzy stanami oraz prawdopodobieństw akceptacji poszczególnych symboli w stanach (na podstawie danych treningowych).

     Pierwsze podejście:

$$a_{ij} = \frac{\text{liczba przejść ze stanu } i \text{ do stanu } j}{\text{liczba wszystkich przejść ze stanu } i}$$

$$b_{jk} = \frac{\text{liczba akceptacji symbolu } k \text{ w stanie } j}{\text{liczba wszystkich akceptacji w stanie } j}$$

PROBLEM: symbole nieznane nie są akceptowane

Jak trenować model? (3)

- Trening przebiega dwuetapowo:
  2. Określenie prawdopodobieństw przejść pomiędzy stanami oraz prawdopodobieństw akceptacji poszczególnych symboli w stanach (na podstawie danych treningowych).

     Drugie podejście (wygładzanie Laplace'a):

     T_{jk} = liczba akceptacji symbolu k w stanie j

     T_j = liczba wszystkich akceptacji w stanie j

$$b_{jk} = \frac{T_{jk} + 1}{T_j + \mathrm{m}}$$

     Inne podejścia: ...

Jak szukać rozwiązania?

- Chcemy symbolom $o_1 o_2 ... o_k$ przyporządkować najbardziej prawdopodobne elementy E_{ik}.
  - Pierwsze podejście:

  Sprawdzamy wszystkie możliwości: $O(n^k)$

  - Drugie podejście:

  Algorytm Viterbi'ego (dynamiczny): $O(kn^2)$

Co gdy zawiodą słowniki?

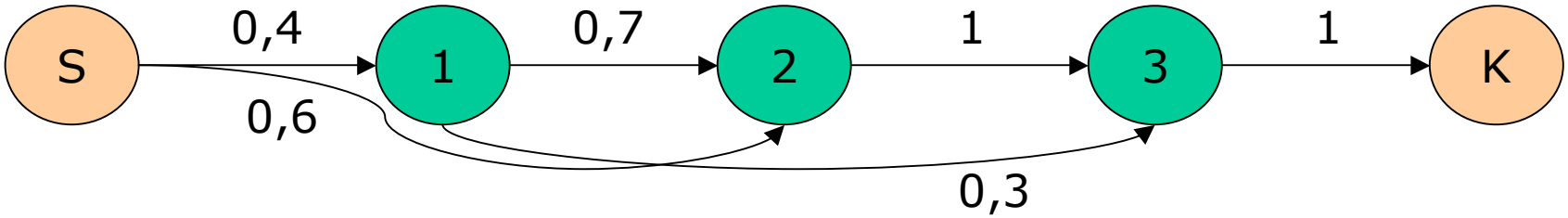
- Słowniki ZAWSZE będą niekompletne
- Nie wszystko da się załatwić manipulując prawdopodobieństwami akceptacji symboli nieznanych

Taksonomia – przykład (1)

3	0,3
45	0,3
66	0,3

A	0,7
C	0,2

B	0,6
C	0,3



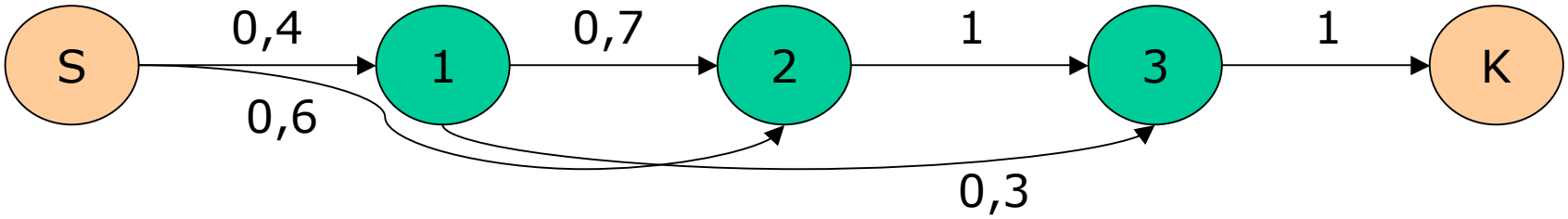
WEJŚCIE: 90#D

Taksonomia – przykład (2)

3	0,3
45	0,3
66	0,3

A	0,7
C	0,2

B	0,6
C	0,3

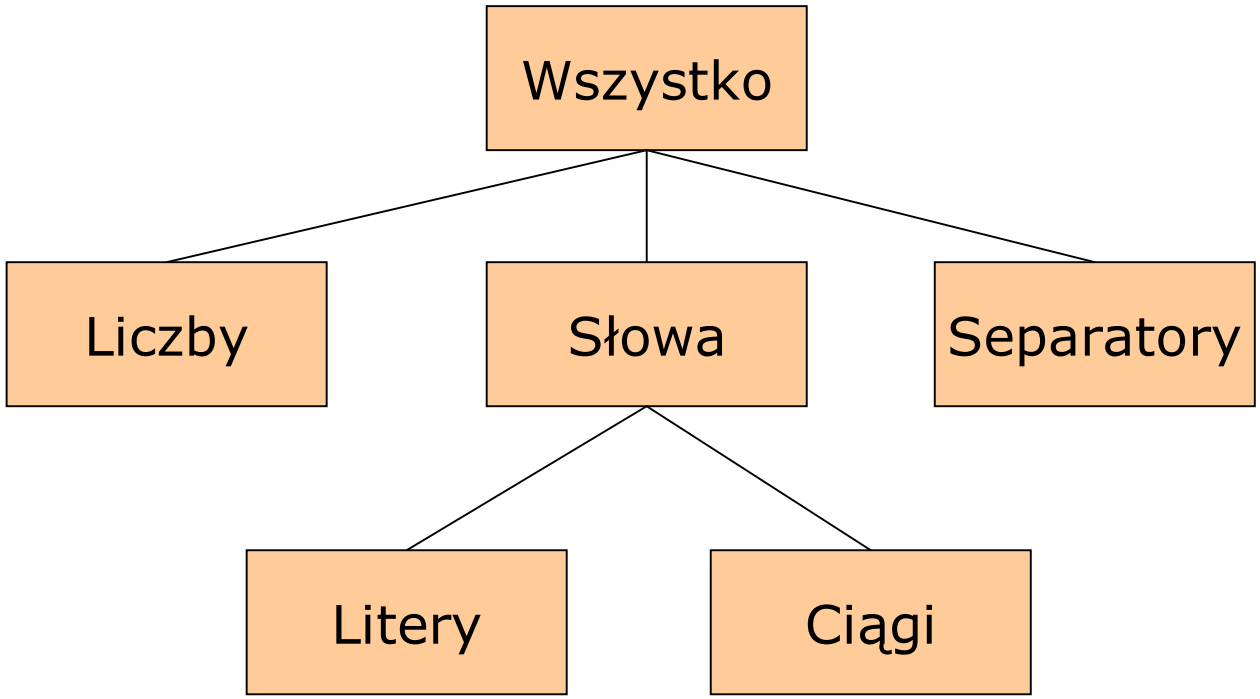


WEJŚCIE: 90#D

WYJŚCIE: <„90”, 2>, <„D”, 3>!!! bo
0,6*0,1*1*0,1 > 0,4*0,1*0,3*0,1

PROBLEM: prawdopodobieństwa dla liczb!

Taksonomia – przykład (3)



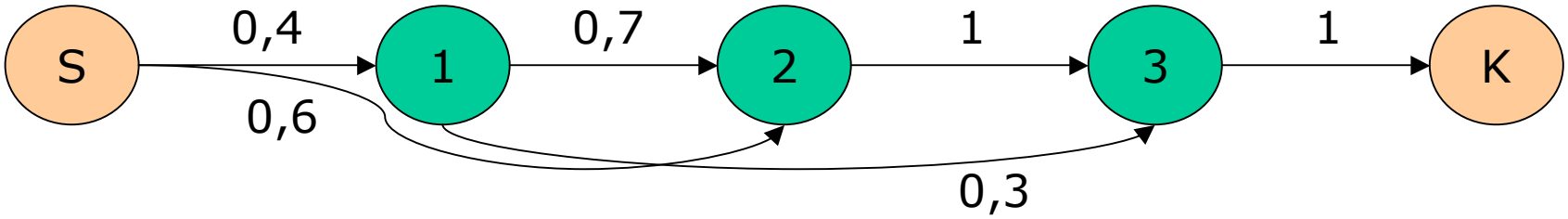
Prawdopodobieństwa określamy jedynie dla liści drzewa taksonomii!

Taksonomia – przykład (4)

LICZBA	0,9

A	0,7
C	0,2

B	0,6
C	0,3



WEJŚCIE: 90#D

WYJŚCIE: <„90”, 1>, <„D”, 3> ☺ bo
0,6*0,1*1*0,1 < 0,4*0,9*0,3*0,1

Co z kolejnością wewnątrz jednego elementu? (1)

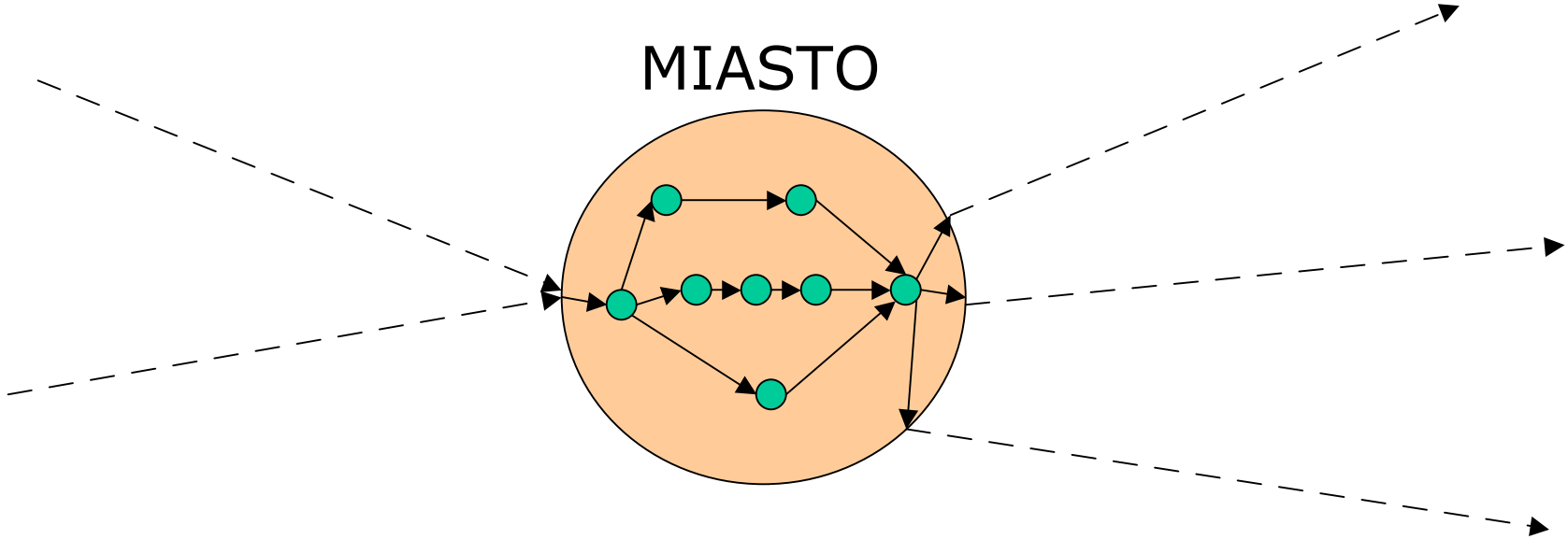
- PROBLEM: Struktura HMM wychwytuje kolejność elementów, ale nie wychwytuje kolejności symboli wewnątrz jednego elementu
  - np.

    Grodzisk Mazowiecki a Mazowiecki Grodzisk

    NEW ZEALAND a ZEALAND NEW

Co z kolejnością wewnątrz jednego elementu? (2)

- Rozwiązanie: stan jako wewnętrzny HMM



  - zwiększa stopień skomplikowania modelu ☹
  - komplikuje fazę treningową ☹

ALE: poprawia skuteczność działania modelu ☺

Rozszerzenia modelu

- Dodatkowa baza danych, np. geograficzna
- <„Rybnik”, MIASTO>, <„śląskie”, WOJEWÓDZTWO>, <???, KRAJ>

  POLSKA czy CZECHY ???

  - Dołączając bazę geograficzną będziemy w stanie odpowiedzieć na to pytanie.
  - Prawdopodobieństwo akceptacji powinno teraz zależeć od tego, co zostało zaakceptowane w stanie poprzedzającym stan bieżący
  – wymaga modyfikacji algorytmu Viterbi’ego ☹

Wyniki

- Dane testowe podzielone ręcznie – dla porównywania wyników
- 99% poprawnych podziałów dla adresów amerykańskich (50 rekordów treningowych / 700 testowych)
- 90% dla adresów azjatyckich
- 87,3% dla danych bibliograficznych (100/205)

- Zwiększanie liczby rekordów treningowych nie poprawia znacząco precyzji podziału!

Pomysł na pracę magisterską

- Aplikacja do tworzenia modeli

  + rozszerzenie o możliwość wyłapywania literówek

    np. Jat Kowalski

  + ...
- Sprawdzenie jak to działa na przykładzie polskich adresów
- ...