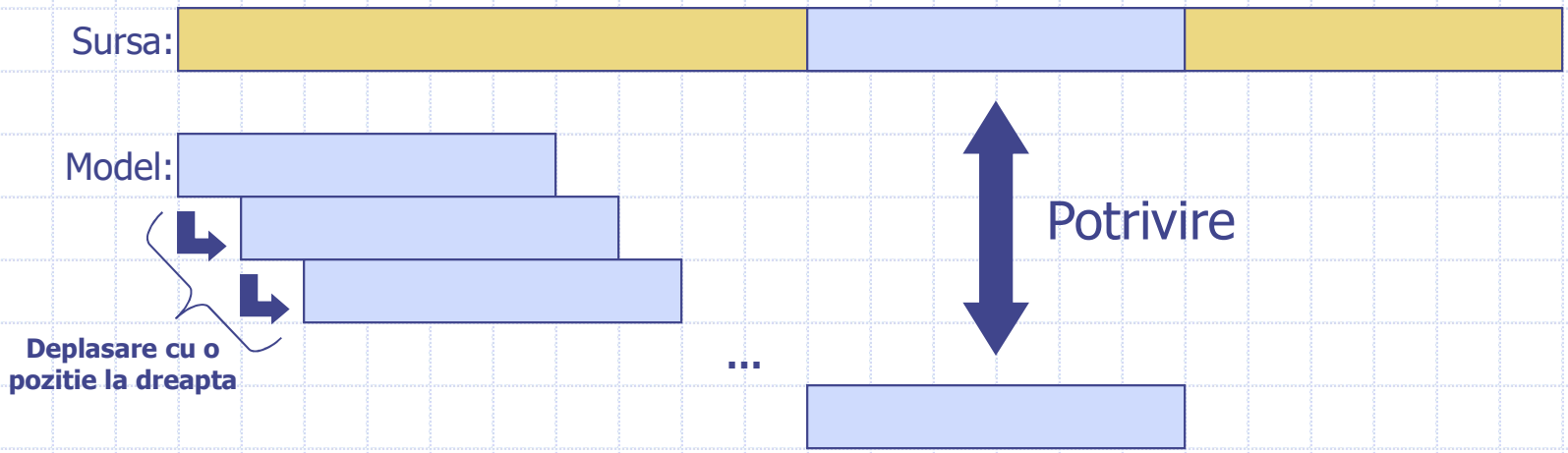
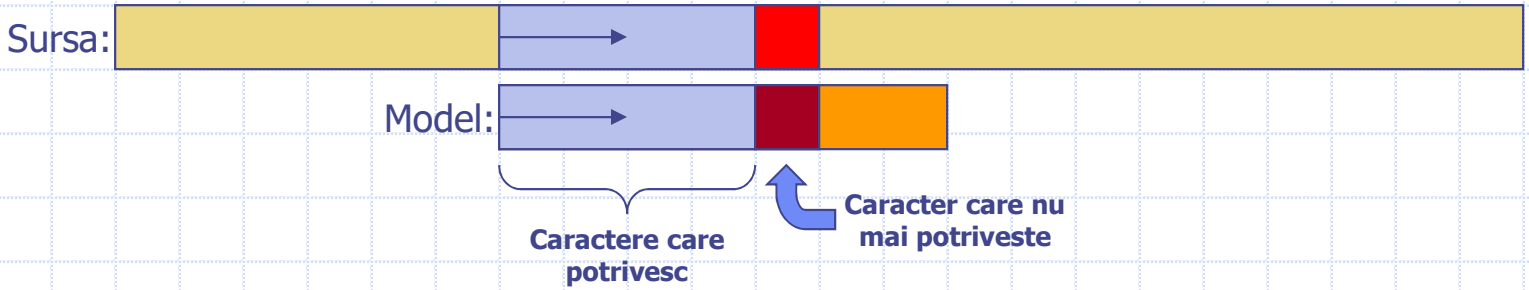
Algoritmul Knuth-Morris-Pratt

- Algoritmul imaginat de Knuth, Morris si Pratt reprezinta o tehnica avansata de cautare a unui sir de caractere model (pattern) intr-un sir de caractere sursa



Algoritmul Knuth-Morris-Pratt

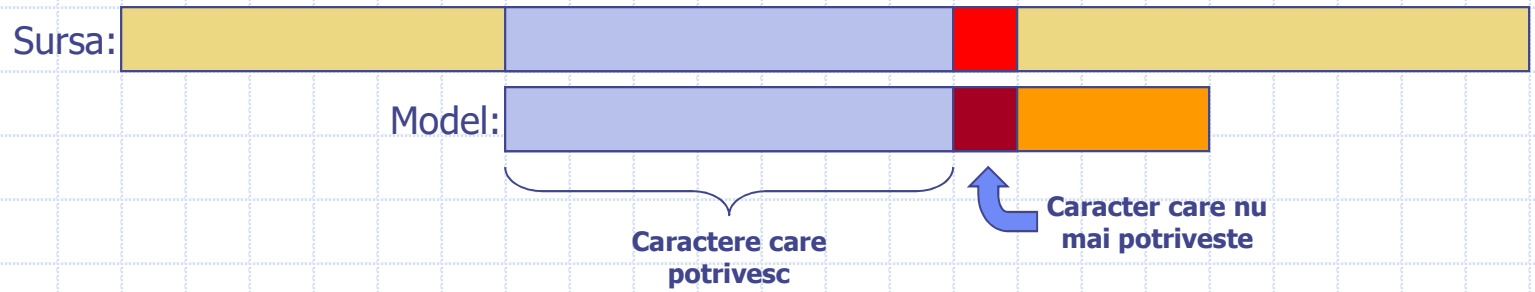
- Spre deosebire de varianta clasica, in care modelul este deplasat de-a lungul sursei cu cate o pozitie pana la gasirea unei potriviri, algoritmul Knuth-Morris-Pratt incearca deplasarea modelului cu un numar mai mare de pozitii in momentul depistarii unei nepotriviri



- In figura de mai sus, in momentul depistarii nepotrivirii, algoritmul detine informatii despre caracterele din portiunea de culoare albastru-deschis, atat din sursa cat si din model
- Aceste informatii ar putea fi folosite pentru mutarea "inteligenta" a modelului spre dreapta cu mai mult de o pozitie

Algoritmul Knuth-Morris-Pratt

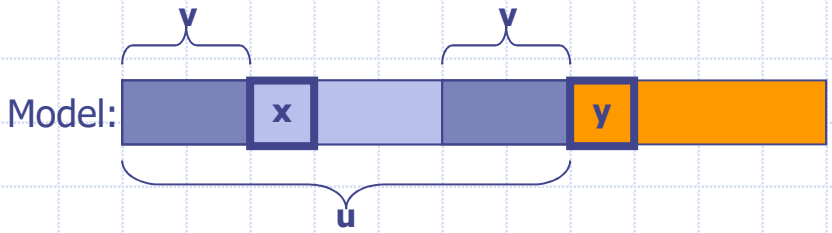
- De fiecare data, in cazul unei nepotriviri intre sursa si model, situatia se va prezenta ca in figura de mai jos:



- Vom nota cu **u** portiunea din model care potriveste cu portiunea corespunzatoare din sursa (adica portiunea colorata in albastru-deschis)
- Acest subsir ar putea fi chiar sirul vid, in cazul in care chiar primul caracter al modelului nu potriveste cu caracterul corespunzator din sursa

Algoritmul Knuth-Morris-Pratt

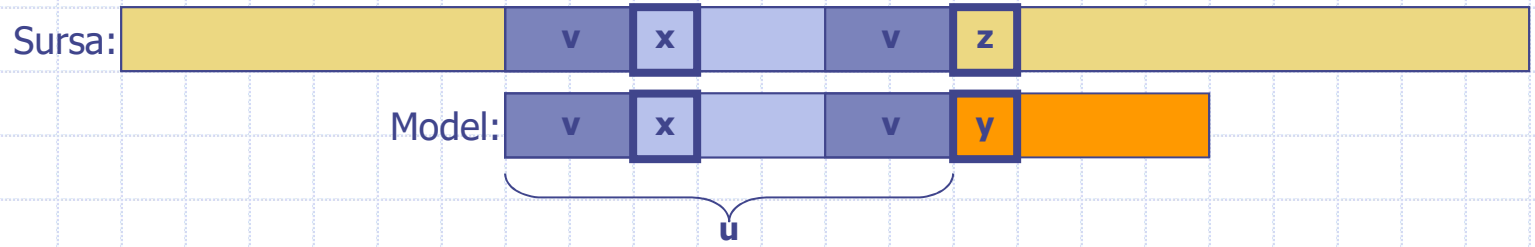
- Este clar ca modelul trebuie deplasat spre dreapta, pentru a incerca o noua potrivire



- In acest moment, se incearca gasirea unui subsir **v** (diferit de **u**) de **lungime maxima** care este atat prefix cat si sufix pentru sirul **u**
- De asemenea, caracterele **x** si **y**, dispuse ca in figura, ***trebuie sa fie diferite***
- Subsirul **v** va fi mai scurt decat **u** – chiar daca **u** este un subsir al lui **u** de lungime maxima care este si sufix si prefix pentru **u**, el nu indeplineste conditia referitoare la **x** si **y**

Algoritmul Knuth-Morris-Pratt

- **x** este caracterul care urmeaza primei aparitii a lui **v** in **u**
- **y** este caracterul care urmeaza ultimei aparitii a lui **v** in **u**, adica este chiar caracterul din model care a cauzat nepotrivirea intre model si sursa



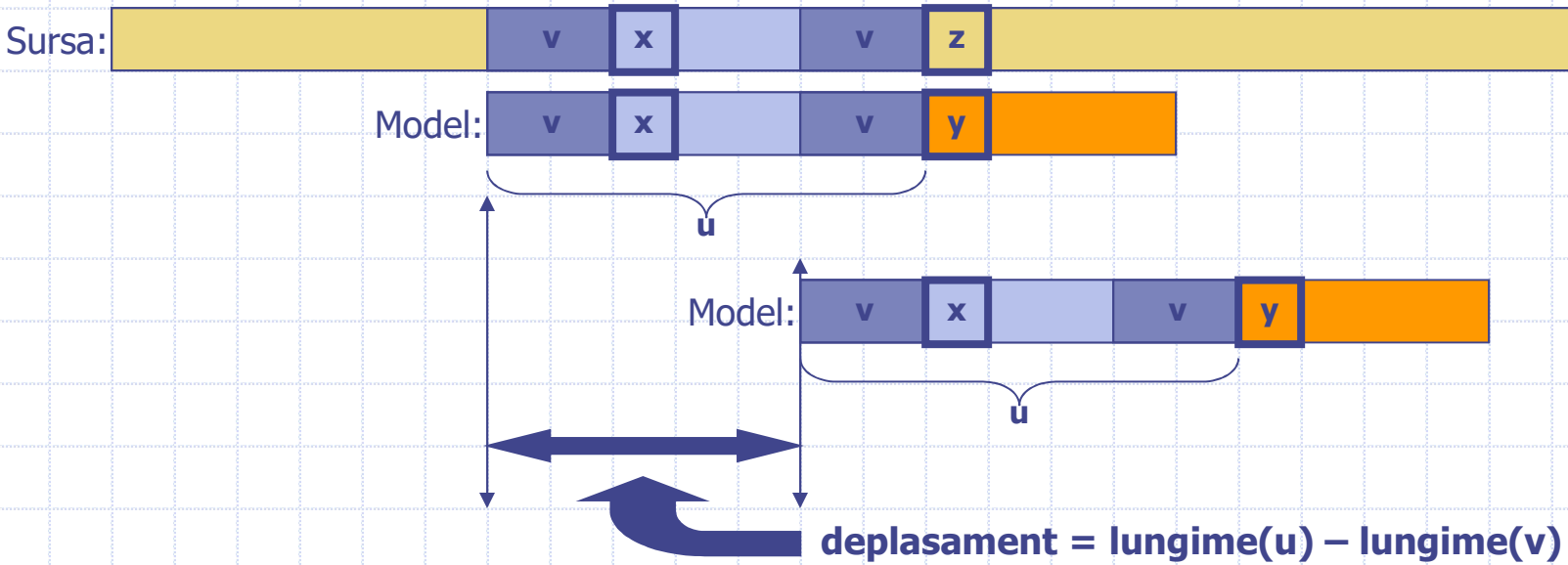
- In momentul acesta cunoastem destule informatii despre caracterele din sursa, in zona in care se afla modelul – aceste informatii ne pot permite sa nu mutam modelul spre dreapta cu o pozitie, ci cu mai multe
- Pe langa informatiile care se vad, mai stim si ca **z** != **y** (nepotrivire)

Algoritmul Knuth-Morris-Pratt

- Apar acum doua situatii, pe care le vom studia separat:
  - daca s-a gasit un subsir **v** (chiar vid), astfel incat **x** != **y**, atunci se memoreaza intr-o variabila **k** lungimea subsirului **v**
  - daca nu s-a gasit un subsir **v** astfel incat **x** != **y**, se memoreaza in variabila **k** valoarea –1
- **Nepotrivirea va cauza o deplasare a modelului spre dreapta cu o cantitate egala cu lungime(u) - k**
- Se observa ca nici una din cele doua situatii nu depinde de caracterele sursei, ci doar de caracterele modelului, ceea ce inseamna ca, in functie de pozitia **j** din model unde apare nepotrivirea, se poate calcula inca de la inceput valoarea **tabDepl[j] = k**
- Avantajul este ca aceasta tabela de deplasari se calculeaza o singura data, la startul algoritmului, si apoi, de cate ori se gaseste o nepotrivire intre sursa si model la pozitia **j** a modelului, se deplaseaza modelul spre dreapta cu un numar de pozitii egal cu **j – k = j – tabDepl[j]**, castigandu-se un timp important
- Se observa ca pozitia **j** la care a aparut nepotrivirea este egala cu lungimea sirului **u** (pozitiile **j** se numara incepand cu 0)

Algoritmul Knuth-Morris-Pratt

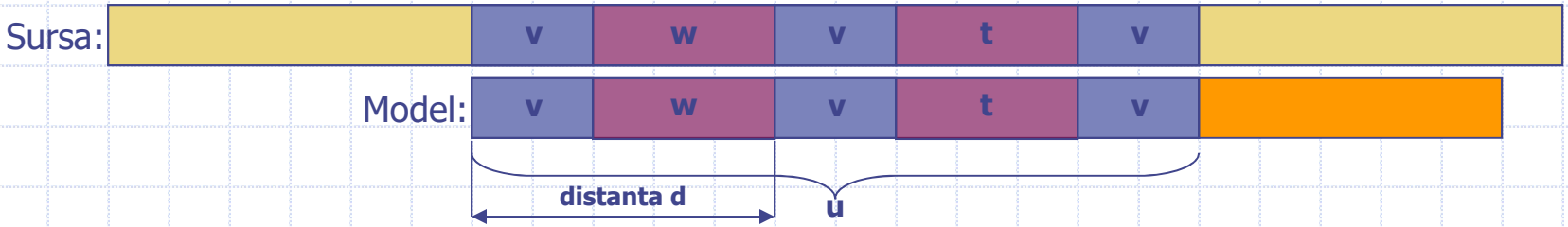
- Am presupus ca **x** != **y** (stim ca **y** != **z**)



- Cum **x** != **y** si **y** != **z**, s-ar putea ca **x** == **z**, deci am deplasat modelul exact deasupra unei portiuni din sursa unde sunt sanse sa gasim o noua potrivire

Algoritmul Knuth-Morris-Pratt

- Justificam cazul in care subsirul **v** mai apare o data in interiorul lui **u**



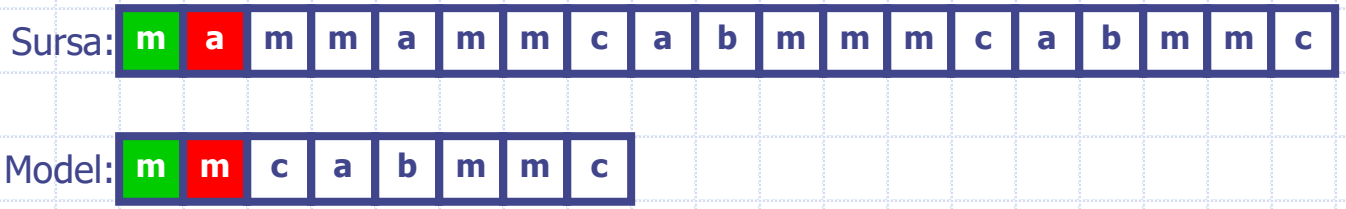
- Pe buna dreptate, se poate pune intrebarea: "In situatia de mai sus, de ce nu se deplaseaza modelul cu mai putine pozitii, anume, doar pe distanta **d** (vezi figura), astfel incat subsirul **v** initial sa ajunga sub al doilea subsir **v** din cadrul lui **u**, deoarece acolo ar putea exista urmatoarea potrivire intre sursa si model"
- Evident, pentru a exista o potrivire acolo, ar trebui ca subsirul notat cu **w** sa fie identic cu subsirul notat cu **t**, deoarece acestea ar ajunge unul sub altul
- Dar, daca **w** este identic cu **t**, atunci inseamna ca nu l-am ales bine pe **v** – acesta nu este cel mai lung subsir al lui **u** care este si prefix si sufix pentru **u** (acest subsir ar fi fost **v** + **w** + **v**)
- Din moment ce l-am calculat bine pe **v**, inseamna ca **w** nu poate fi identic cu **t**, deci potrivirea de care am vorbit nu se poate manifesta

Algoritmul Knuth-Morris-Pratt

- Pentru cazul prezentat anterior, am demonstrat ca odata ce **v** a fost bine calculat, nu conteaza de cate ori apare acesta in interiorul lui **u** si ca o deplasare a modelului cu cel putin lungime(**u**) – lungime(**v**) nu duce la pierderi de solutii
- Doar aparitia lui **v** de la sfarsitul lui **u** conteaza
- Evident, situatia de pe slide-ul anterior nu acopera cazul in care subsirul **v** apare de mai multe ori in cadrul lui **u** si aceste aparitii se intercaleaza
- Si pentru astfel de situatii, se poate demonstra, folosind un rationament asemanator, ca nu este cazul unei deplasari mai scurte decat lungime(**u**) – lungime(**v**) a modelului
- Demonstratia acestui fapt se propune cititorului ca exercitiu

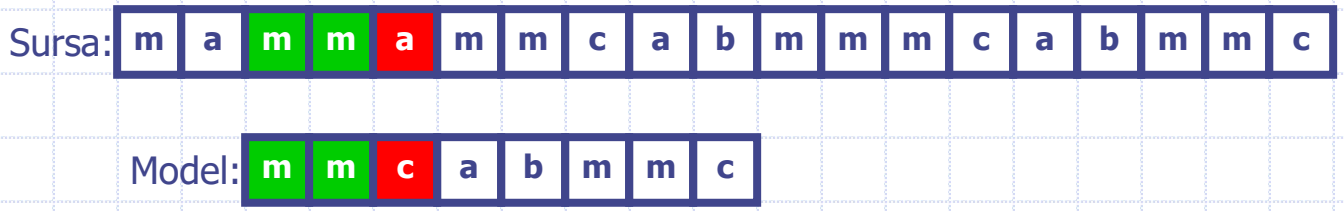
Algoritmul Knuth-Morris-Pratt

- Vom studia mersul algoritmului pentru urmatoarea situatie:



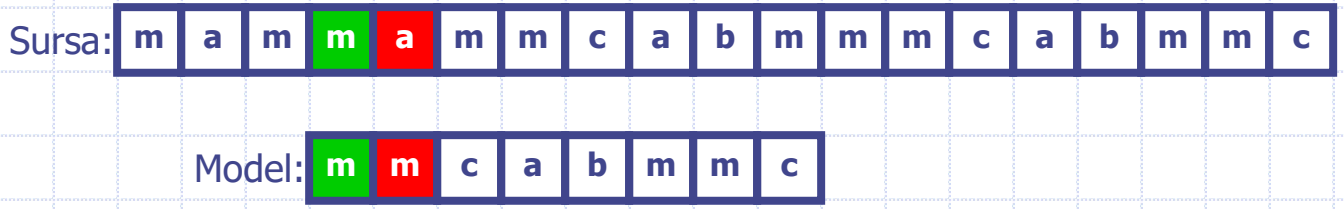
- Prima nepotrivire apare intre caracterele ‘a’ si ‘m’
- Sirul **u** = “m”
- Sirul **v** = “” (sirul **v** nu poate fi egal cu **u**)
- **x** = ‘m’ si **y** = ‘m’ (**x** == **y**)
- Cum **x** == **y** si nu avem alta varianta pentru **v**, rezulta **k** = -1
- Algoritmul cere o deplasare a modelului spre dreapta cu lungime(**u**) – **k** = 2

Algoritmul Knuth-Morris-Pratt



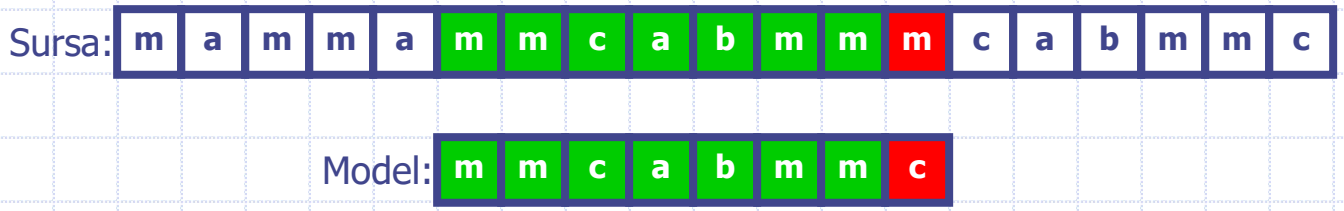
- Prima nepotrivire apare intre caracterele ‘a’ si ‘c’
- Sirul **u** = “mm”
- Sirul **v** = “m”
- **x** = ‘m’ si **y** = ‘c’ - este OK, **x** != **y**
- **k** = lungime(**v**) = 1
- Algoritmul cere o deplasare spre dreapta a modelului cu lungime(**u**) – **k** = 1

Algoritmul Knuth-Morris-Pratt



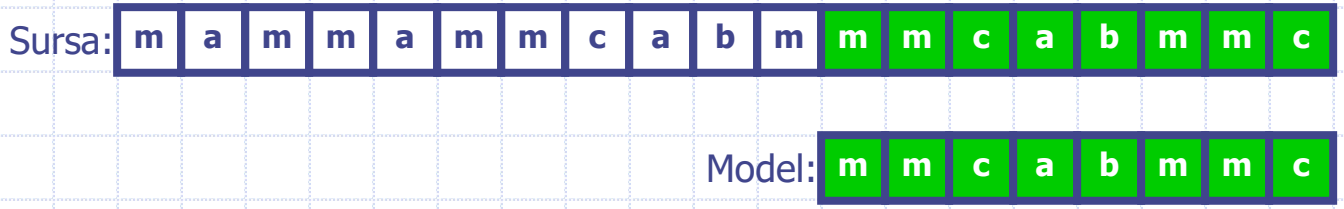
- Prima nepotrivire apare intre caracterele ‘a’ si ‘m’
- Sirul **u** = “m”
- Sirul **v** = “”
- **x** = ‘m’ si **y** = ‘m’ (**x** == **y**)
- Cum **x** == **y** si nu avem alta varianta pentru **v**, rezulta **k** = -1
- Algoritmul cere o deplasare a modelului spre dreapta cu lungime(**u**) – **k** = 2
- Anticipand, putem observa ca de fiecare data cand nepotrivirea apare la caracterul al doilea al modelului, acesta trebuie deplasat cu 2 caractere spre dreapta, independent de sursa
- Aceste deplasari pot fi calculate a priori si tabelate pentru fiecare posibila pozitie a nepotrivirii, astfel incat sirul **v** sa nu mai trebuiasca calculat de fiecare data, ci deplasarea sa poata fi scoasa direct din tabel

Algoritmul Knuth-Morris-Pratt



- Prima nepotrivire apare intre caracterele ‘m’ si ‘c’
- Sirul **u** = “mmcabmm”
- Daca alegem sirul **v** = “mm”, ar rezulta **x** = ‘c’ si **y** = ‘c’ (**x** trebuie sa fie diferit de **y**)
- Alegem urmatoarea varianta pentru **v**, anume **v** = “m”
- **x** = ‘m’ si **y** = ‘c’ – este OK, **x** != **y**
- **k** = lungime(**v**) = 1
- Algoritmul cere o deplasare a modelului spre dreapta cu lungime(**u**) – **k** = 6

Algoritmul Knuth-Morris-Pratt



- S-a gasit o potrivire intre sursa si model
- In acest moment, algoritmul se poate incheia, sau poate continua pentru gasirea altor potriviri
- Daca se decide continuarea, se poate considera ca **u** este egal cu intreg modelul, **v** se calculeaza si se deplaseaza modelul spre dreapta cu lungime(**u**) – lungime(**v**) (deoarece nu dispunem de caracterul **y**)
- In cazul prezentat, continuarea nu va duce la gasirea altor potriviri, deoarece am ajuns la sfarsitul sirului sursa

Algoritmul Knuth-Morris-Pratt

- Daca tabloul de deplasari in caz de nepotrivire (**tabDepl**) este calculat a priori, performanta algoritmului ajunge la O(M+N), unde M este lungimea sirului model iar N este lungimea sirului sursa
- Practic, tabloul de deplasari nu depinde de sursa, ci doar de model
- **tabDepl[j]** va contine valorile de tip **k** calculate asa cum se specifica pe unul din slide-urile anterioare
- Mai departe, de cate ori apare o nepotrivire intre sursa si model la pozitia **j** din model, acesta se va deplasa cu **j – tabDepl[j]**
- Pentru exemplul considerat, construirea tabloului de deplasari se propune ca exercitiu
- Rezultatul trebuie sa fie tabDepl:

-1	-1	1	0	0	-1	-1	1