## Equivalentie van DFA en 2-DFA

- definitie van 2-DFA
- een equivalentierelatie gebaseerd op een 2-DFA voor een L
- het is een MN(L)-relatie !

Snel opgeschreven op 20 november 2011 door Bart Demoen

**Definitie van 2-DFA**

Informeel: 2-DFA kan leeskop links en rechts bewegen ...

- $(Q,\ \Sigma,\ \delta,\ \vdash,\ \dashv,\ q_s,\ q_a,\ q_r)$
  - $\vdash$ en $\dashv$ niet in $\Sigma$ (linkse en rechtse marker)
  - $\delta$: $(Q \times (\Sigma \cup \{\vdash, \dashv\})) \to Q \times \{L, R\}$
  - $\{q_s, q_a, q_r\} \subseteq Q$ (start, accept en reject toestand)
  - $\#\{q_s, q_a, q_r\} = 3$
- niet te links of te rechts lezen ...
  - $\forall p, \exists q : \delta(p, \vdash) = (q \times R)$ en $\forall p, \exists q : \delta(p, \dashv) = (q \times L)$
- bij accept/reject: leeskop helemaal naar rechts
  - $\delta(q_a, \_) = (p \times R)$ en $\delta(q_r, \_) = (p \times R)$

**String s geaccepteerd door 2-DFA**

- start 2-DFA in $q_s$ op string $\vdash s \dashv$
- geaccepteerd als de 2-DFA in de situatie $(q_a, \dashv)$ komt

  s behoort tot de taal van de 2-DFA
- verworpen als de 2-DFA in de situatie $(q_r, \dashv)$ komt

  s behoort niet tot de taal van de 2-DFA
- anders ...

  s behoort niet tot de taal van de 2-DFA

2-DFA mag in een lus gaan en is dus een beetje zoals een TM

strings waarvoor de 2-DFA in een lus gaat behoren niet tot de taal bepaald door de 2-DFA

**Overtuig jezelf ...**

dat de *speciale* eisen niet *belangrijk* zijn (wat betekent dat juist in deze context ?)

maar ze zullen het ons wel iets gemakkelijker maken

**Het plan**

Gegeven een 2-DFA die taal L bepaalt:

- we definieren een equivalentierelatie $\equiv$ op $\Sigma^*$
- we bewijzen dat $\equiv$ een MN(L)-relatie is
- dat bewijst dat L regulier is

vanaf nu is de 2-DFA (en L) gegeven ...

## Voorbereiding van $\equiv$ - deel I

Voor een gegeven string $s$ maken we een functie

$$T_s : (Q \cup \bullet) \to (Q \cup \perp)$$

$T_s(\bullet)$

- start 2-DFA op $\vdash s \dashv$ met leeskop op $\vdash$ in $q_s$
- de eerste keer dat de leeskop op $\dashv$ komt, is de 2-DFA in toestand $q$, dan $T_s(\bullet) \triangleq q$
- als er zo geen eerste keer is: $T_s(\bullet) \triangleq \perp$

$T_s(p)$

- start 2-DFA op $\vdash s \dashv$ met leeskop op symbool vlak voor $\dashv$ in toestand $p$
- de eerste keer dat de leeskop daarna op $\dashv$ komt, is de 2-DFA in toestand $q$, dan $T_s(p) \triangleq q$
- als er zo geen eerste keer is: $T_s(p) \triangleq \perp$

## Voorbereiding van $\equiv$: deel II

Wat betekent die $T_s$ ?

- volg de werking van de 2-DFA op een string van de vorm $sw$
- elke keer dat de leeskop van $w$ naar $s$ gaat en daar aankomt in toestand $p$
- dan: de eerste keer dat de leeskop daarna van $s$ naar $w$ gaat, komt ie daar toe in $T_s(p)$ tenzij dat nooit gebeurt
- plus een speciaal geval voor $T_s(\bullet)$ want dat is een overgang van $s$ naar $w$ waarvoor geen vorige overgang van $w$ naar $s$ was

$T_s$ bevat alle info die van $s$ naar $w$ kan vloeien

**Interludium over $T_s$**

- als je wil beslissen of string $sw$ geaccepteerd wordt, dan heb je genoeg aan
  - $T_s$
  - $w$
  - $\delta$
- maar $s$ moet je niet kennen

Lees niet verder totdat je dat verstaat !

Hint: speel zelf voor 2-DFA en kijk wat je nodig hebt als die een overgang wil maken van $w$ naar $s$

## Voorbereiding van $\equiv$: deel III

hoeveel verschillende functies $(Q \cup \bullet) \to (Q \cup \perp)$ bestaan er ?

- stel $\#Q = k$, dan $(k+1)^{(k+1)}$
- juiste aantal niet belangrijk, wel dat het eindig is !

**Definitie van $\equiv$ op $\Sigma^*$**

- $s \equiv t \Leftrightarrow T_s = T_t$
- bewijs zelf dat dit een equivalentierelatie is
- de geïnduceerde partitie is eindig, want ...
- als we nu nog kunnen bewijzen dat
  - die partitie $\{L, \overline{L}\}$ verfijnt en
  - dat $\equiv$ rechts-congruent is ...

  **Dan is $L$ regulier !**

## $\equiv$ verfijnt $\{L, \overline{L}\}$: bewijs

Stel $T_s = T_t$ en $s \in L$

- de overgangen van $s$ naar $\dashv$ en die van $t$ naar $\dashv$ gebeuren in dezelfde toestanden en in dezelfde volgorde (waarom ?)
- laatste overgang voor $s$ is in toestand $q_a$
- dus ook voor $t$

dus ook $t \in L$

M.a.w. $T_s = T_t \Leftrightarrow (s \in L) \leftrightarrow (t \in L)$

of

$$\equiv \text{ verfijnt } \{L, \overline{L}\}$$

$\blacksquare$

De Interludium slide voordien kan je ook helpen ...

## $\equiv$ is rechts-congruent: bewijs

Stel $s \equiv t$, dan $T_s = T_t$ (per definitie)

Neem een $a \in \Sigma$

- $T_{sa}(\bullet)$ ($T_{ta}(\bullet)$) krijg je door
  - stel $p = T_s(\bullet)$ $(= T_t(\bullet))$
  - of: de 2-DFA komt de $1^{ste}$ keer met leeskop boven de $a$ in $p$
  - $T_s = T_t$ en de overgangen tussen $s$ en $a$ gebeuren in dezelfde volgorde als tussen $t$ en $a$, dus $T_{sa}(\bullet) = T_{ta}(\bullet)$
- start 2-DFA met leeskop onder $a$ en in een willekeurige toestand $p$
  - leeskop gaat naar rechts ... $T_{sa}(p) = T_{ta}(p)$
  - of leeskop gaat naar links ... dus $T_{sa}(p) = T_{ta}(p)$

$\blacksquare$

**Besluit**

- elke taal L die bepaald wordt door een 2-DFA is regulier
- elke taal L die bepaald wordt door een DFA wordt ook door een 2-DFA bepaald
- 2-DFA en DFA zijn even *krachtig*

**Om over na te denken ...**

- kan een 2-DFA voor een L kleiner zijn dan de minimale DFA ?
- bestaat een minimalisatie-algoritme voor 2-DFA's ?
- hoe zit het met een transformatie van 2-DFA naar DFA ?
- bestaan er ook 2-NFA's ? en welke eigenschappen hebben die ?
- is $H_{2-DFA}$ herkenbaar, beslisbaar ?

**Geschiedenis**

- probleem eerst bestudeerd (en opgelost) door M. Rabin en D. Scott (1959): ingewikkeld bewijs
- J. Shepherdson gaf al snel een eenvoudiger bewijs: het staat essentieel hierboven - Rabin en Scott vinden het zelfs niet de moeite om hun eigen bewijs in detail te publiceren, maar verwijzen naar Shepherdson
- 1989: M. Vardi geeft een nog *gemakkelijker* bewijs - ik vind het moeilijker :-(
- zie boek *Automata and Computability* van Dexter C. Kozen