

Sulundid Elus

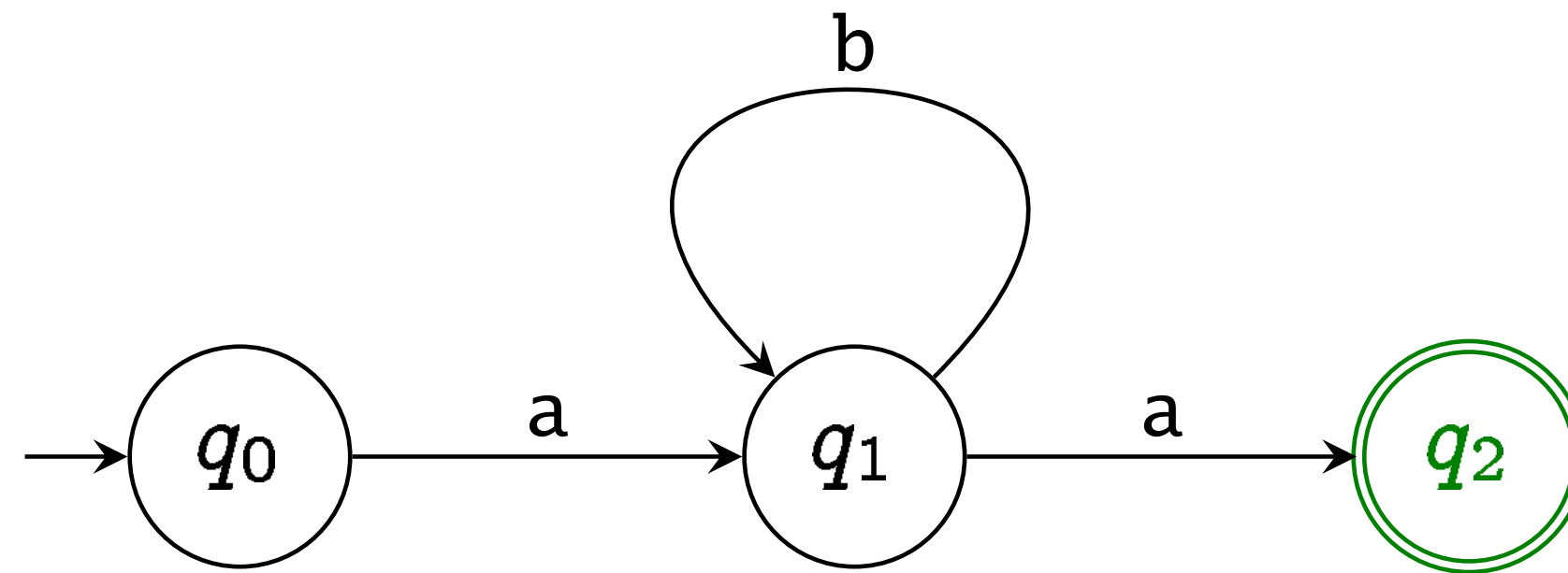
Vesal Vojdani
(TÜ Arvutiteaduse Instituut)

Tagasiside!

- Loeme kõike... (isegi kui me seda ignoreerime.)
- Tegelikult me ei ignoreeri, aga teil on väga erinevad nõudmised!
- Sügavamalt aru saamine komponeerib: **Proge** ◦ **00P** ◦ **A&A**
kuni tasemed on siin nii seinast seinna erinevad...

Kodutöö: NFA

- Järjekordne kordamisülesanne... :)
- Sisuliselt graafi ülesanne.
- Kui see ei tundu nagu lihtsalt kordamine, siis palun üritage natuke mõelda selle üle, miks iseseisvalt ei peaks saama?
- Selle nädala praktikumides alustame sellega, aga siin...



Lõplik automaat $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ **aktsepteerib** keele

$$L(A) = \{w \in \Sigma^* \mid (q_0, w, q_f) \in \delta^*, q_f \in F\}$$

kus $\delta^* \subseteq Q \times \Sigma^* \times Q$ on üleminekurelatsiooni δ refleksiivne transitiivne sulund.

NFA Definiitsioon

Mis on üleminekurelatsiooni refleksiivne transitiivne sulund?

Transitiivsus & Reflektiivsus

- Transitiivsus: kui $(x,y) \in \delta$ ja $(y,z) \in R$, siis ka $(x,z) \in \delta$
- Reflektiivsus: kõik $(x,x) \in \delta$

Transitiivsus & Reflektiivsus

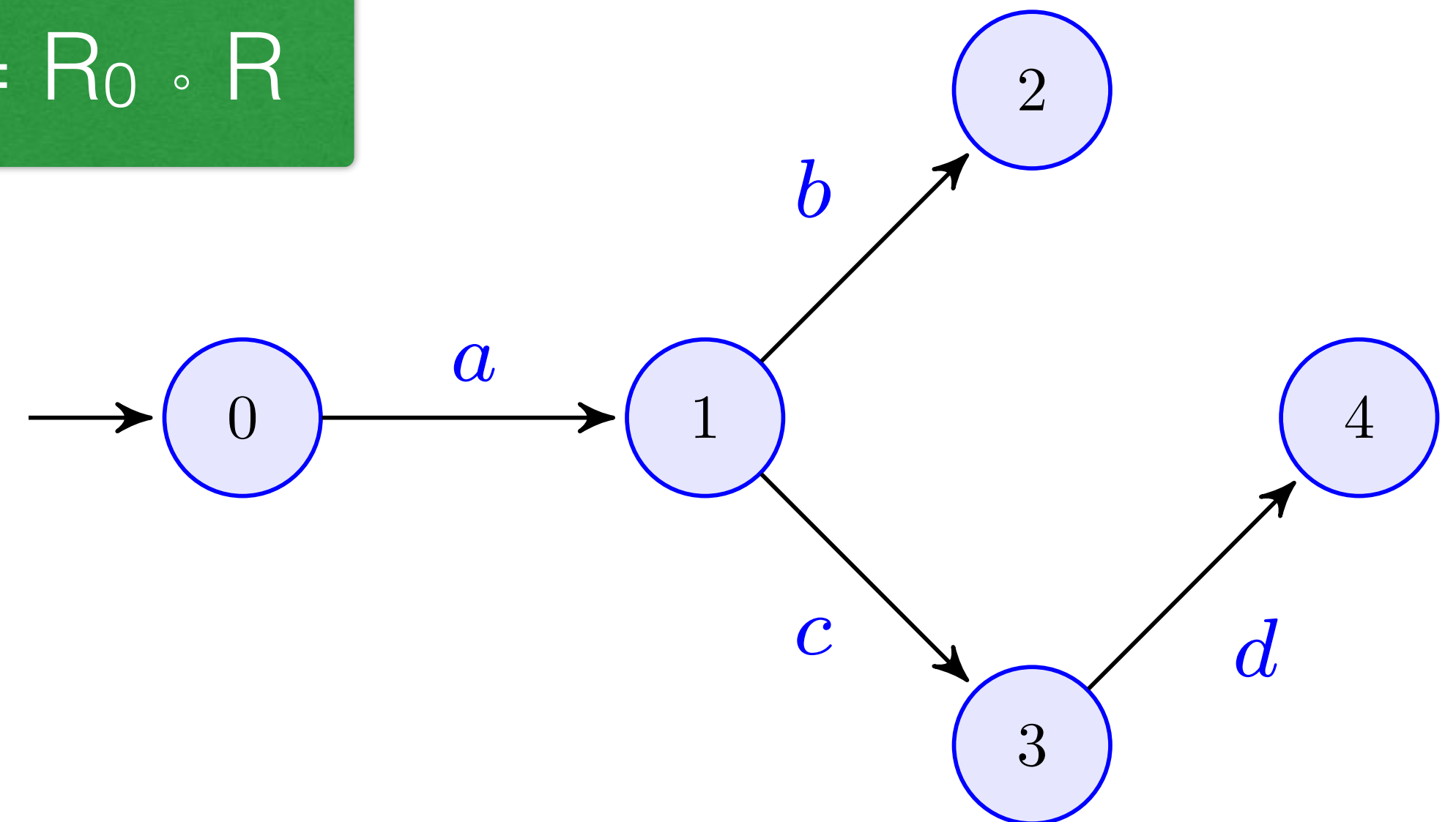
- Transitiivsus: kui $(x, "a", y) \in \delta$ ja $(y, "b", z) \in \delta$, siis ka $(x, "ab", z) \in \delta$
- Reflektiivsus: kõik $(x, \epsilon, x) \in \delta$

Transitiivse sulundi näide

- $(0, a, 1), (1, b, 2), (1, c, 3), (3, d, 4)$ $R_0 = R$

- $(0, ab, 2), (0, ac, 3), (1, cd, 4)$ $R_1 = R_0 \circ R$

- $(0, acd, 4)$ $R_2 = R_1 \circ R$



Relatsioon: Set<Trans>

```
public class Trans {  
    private final int source;  
    private final String label;  
    private final int target;  
    . . .  
}
```


Relatsioonide kompositsioon

```
def compose(rel1: Set[Trans], rel2: Set[Trans]) =  
  for {  
    (from1, s1, dest1) <- rel1  
    (from2, s2, dest2) <- rel2  
    if dest1 == from2  
  } yield (from1, s1 + s2, dest2)
```

Kahjuks me siin Scalas
ei programmeeri...

Relatsioonide kompositsioon

```
public static Set<Trans> compose(Set<Trans> rel1, Set<Trans> rel2) {  
    Set<Trans> result = new HashSet<>();  
    for (Trans t1 : rel1) {  
        for (Trans t2 : rel2) {  
            if (t1.getTarget() == t2.getSource()) {  
                String label = t1.getLabel() + t2.getLabel();  
                result.add(new Trans(t1.getSource(), label, t2.getTarget()));  
            }  
        }  
    }  
    return result;  
}
```

Meil on nüüd funktsioon f



rel on
konstantne!

- Meil on funktsioon $f(x) = \text{compose}(x, rel)$
- Ja tahame lihtsalt korduvalt rakendada: $f(f(\dots f(f(x))))$
- Kuni võib-olla ükskord asi stabiliseerub.
- Kuidas implementeerida?

Sulundi arvuatmine

```
static Set<Trans> transitiveClosure(Set<Trans> rel) {  
    Set<Trans> next = new HashSet<>(rel);  
    Set<Trans> newTrans = compose(next, rel);  
    while (!next.containsAll(newTrans)) {  
        next.addAll(newTrans);  
        newTrans = compose(next, rel);  
    }  
    return next;  
}
```

See kood peaks väljendama
"rakenda compose kuni hulk
enam ei muutu"??

Funktsiooni f püsipunkt

```
fixpoint f x =
```

```
  if f x == x then x else fixpoint f (f x)
```

Kahjuks me siin Haskellis
ei programmeeri...

Funktsiooni f püsipunkt

```
public static <T> T fixpoint(Function<T, T> f, T x) {  
    if (f.apply(x).equals(x)) return x;  
    else return fixpoint(f, f.apply(x));  
}
```

Sulundi arvutamisel alles hoida...

```
public static <T> Set<T> closure(  
    Function<Set<T>, Set<T>> f,  
    Set<T> m) {  
    return fixpoint(x -> union(m, f.apply(x)), m);  
}
```

Ja nüüd saame

```
public static Set<Trans> transitiveClosure(Set<Trans> rel) {  
    return closure(x -> compose(x, rel), rel);  
}
```


Sellega saab ka...

- Delfiine päästa
- Maailma rahu...
- Kodutöö jaoks vajalik epsilon-sulund!