Trojúhelníkové sítě

© **2009-2012 Josef Pelikán, CGG MFF UK Praha**

http://cgg.mff.cuni.cz/~pepca/

pepca@cgg.mff.cuni.cz

Trojúhelníkové sítě (tri-mesh)

- úsporné reprezentace v paměti
  - uložení topologie a incidence
  - „Corner table“
- komprese (EdgeBreaker)
  - ukládání na server, přenos po síti

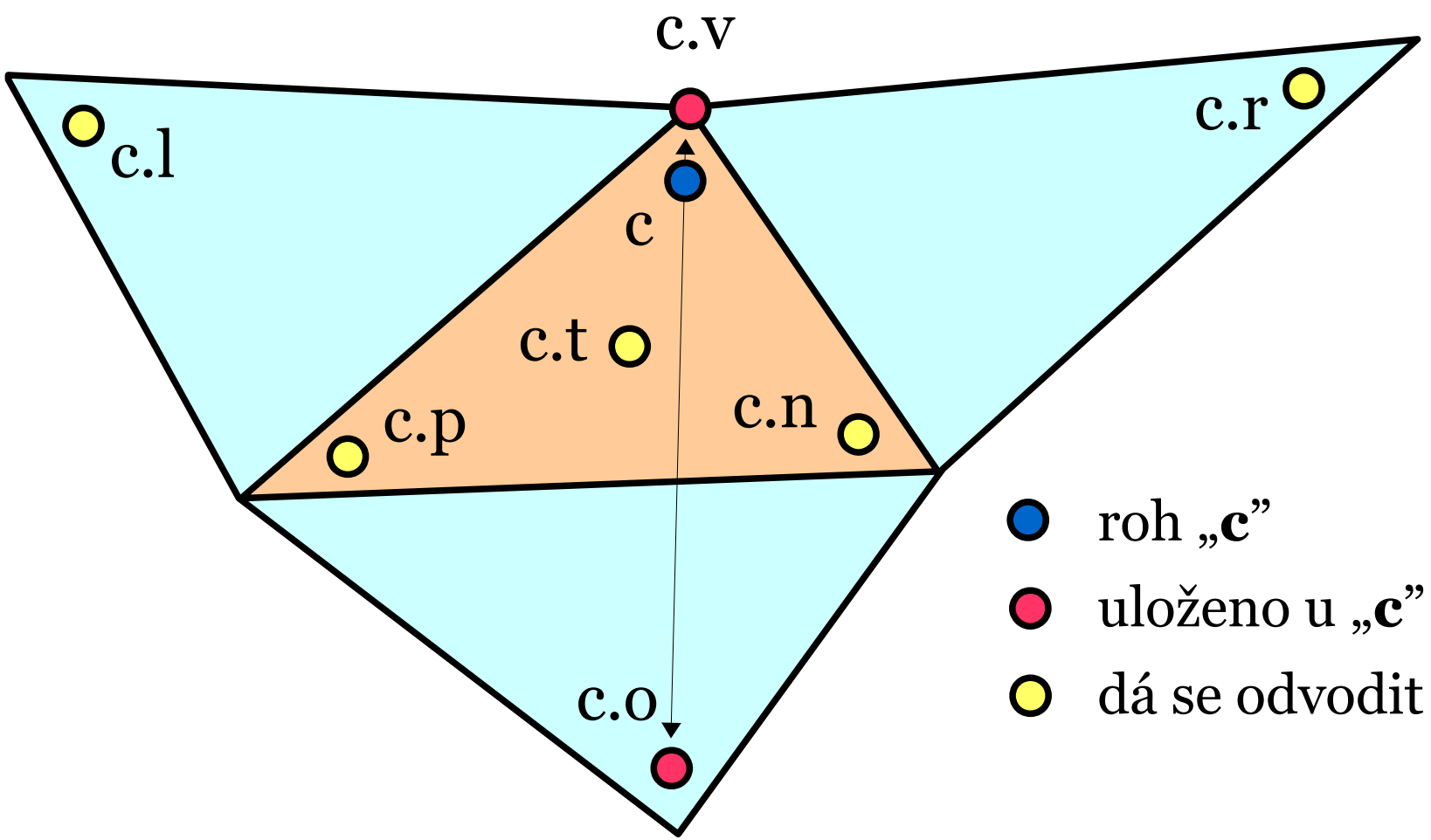
Reprezentace tri-mesh

- podmíněné **HW zobrazováním**
  - pole vrcholů („vertex-array“), „index-array“
  - chybí topologie (informace o sousedech)
- klasická **topologické rozšíření**
  - okřídlená hrana („winged-edge“), půlhrana („half-edge“)
  - implementace s ukazateli je paměťově náročná
- úsporné reprezentace
  - k poli vrcholů přidávají jen minimální informaci
  - např. „**Corner Table**“

„Corner Table“

- tabulka vrcholů G[v]
  - souřadnice, normála, barva, texturové souřadnice, …
- tabulka rohů V[c]
  - jeden vnitřní roh trojúhelníka
  - **index vrcholu** („**c.v**“)
  - rohy jsou uloženy za sebou v 1D poli, CW orientace stěn
  - **protější roh protějšího trojúhelníka** („**c.o**“)
  - implicitní údaje:
    - číslo trojúhelníka **t** = c div 3
    - ostatní rohy **c.n** = (c mod 3 == 2) ? c-2 : c+1, **c.p** = c.n.n
    - další sousední trojúhelníky **c.l** = c.n.o, **c.r** = c.p.o

„Corner Table“



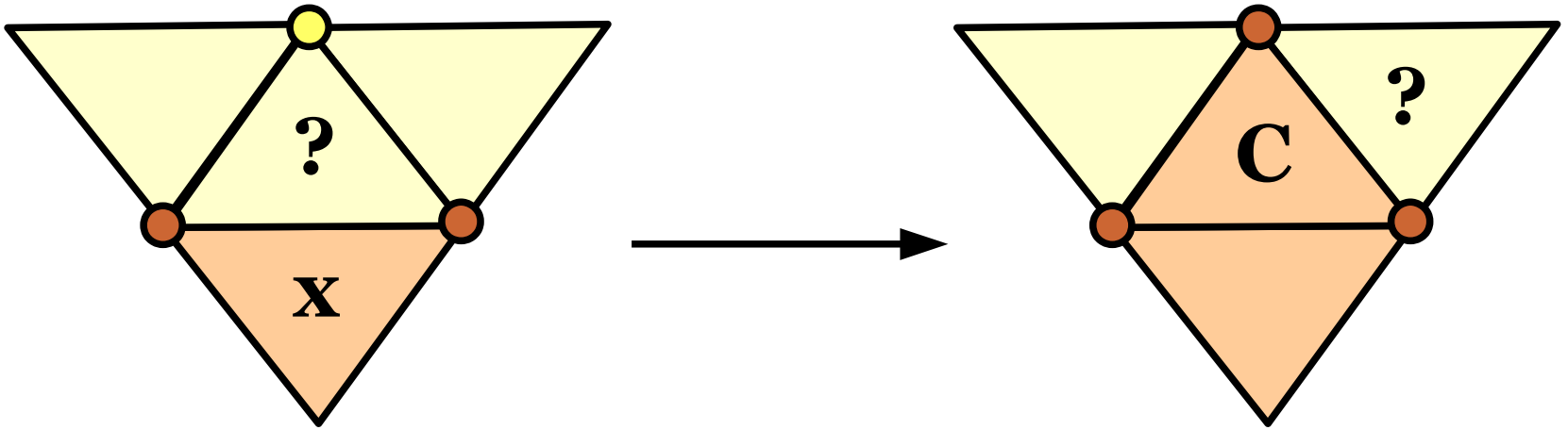
„Corner Table“ – přenos

- tabulka „**o**“ se nemusí přenášet
  - lze ji jednoznačně rekonstruovat z „**v**“
- hodnoty indexů „**v**“ se mohou bitově ořezat
  - index vrcholu obsahuje $31 - \log_2 v$ úvodních nul
- ořezání souřadnic
  - obalový kvádr celého objektu, uvnitř se používají relativní souřadnice (10 až 16 bitů na složku)
  - predikce polohy vrcholů – při inkrementálním průchodu daty (např. „Edgebreaker“) – další úspory

„Edgebreaker“ (Rossignac, 1999)

- úsporné kódování tri-mesh pomocí inkrementálního průchodu sítí
  - **pozice vrcholů** se komprimují za pomoci predikce (až 7 bpv)
  - **topologie sítě** se ukládá velice úsporně na základě průchodu (1.0 až 1.8 bpv)
- „CLERS“ pole
  - **5 možností**, jak pokračovat z aktuálního trojúhelníka
  - možnost entropické komprese (některé kroky/ posloupnosti jsou mnohem pravděpodobnější)

CLERS kódování – C



x předchozí trojúhelník

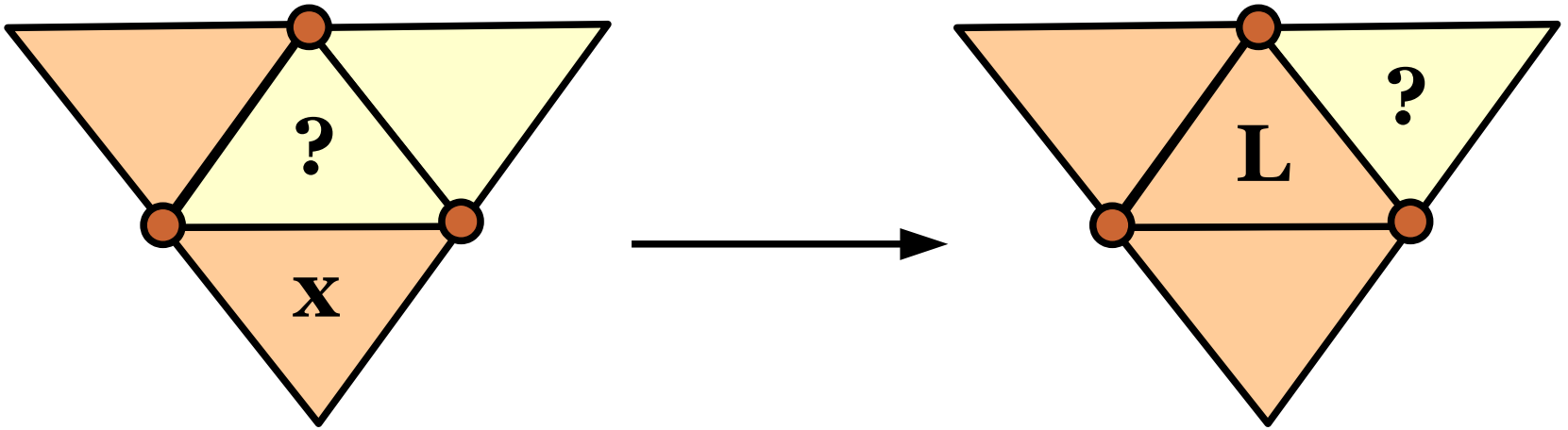
? aktuální trojúhelník

navštívený vrchol

nenavštívený vrchol

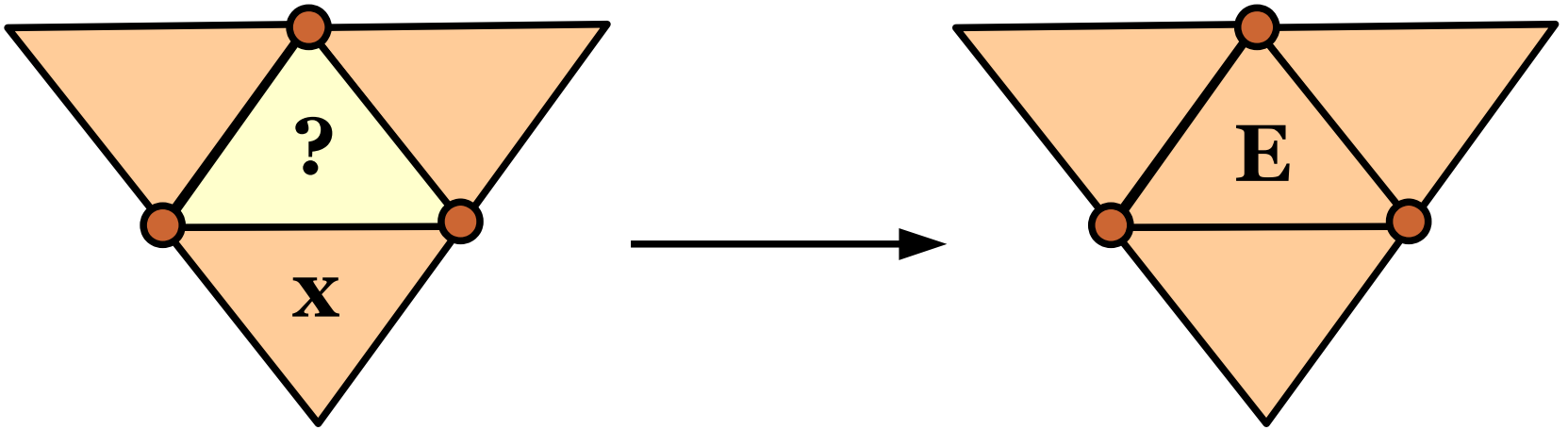
```
write( "C" );
c = c.r;
```

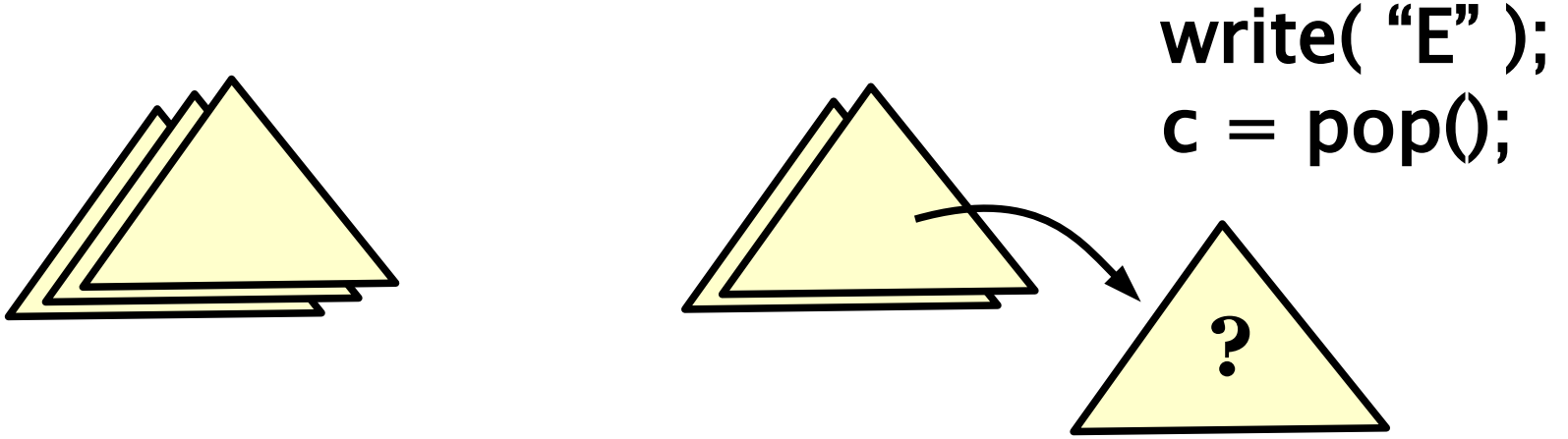
CLERS kódování – L



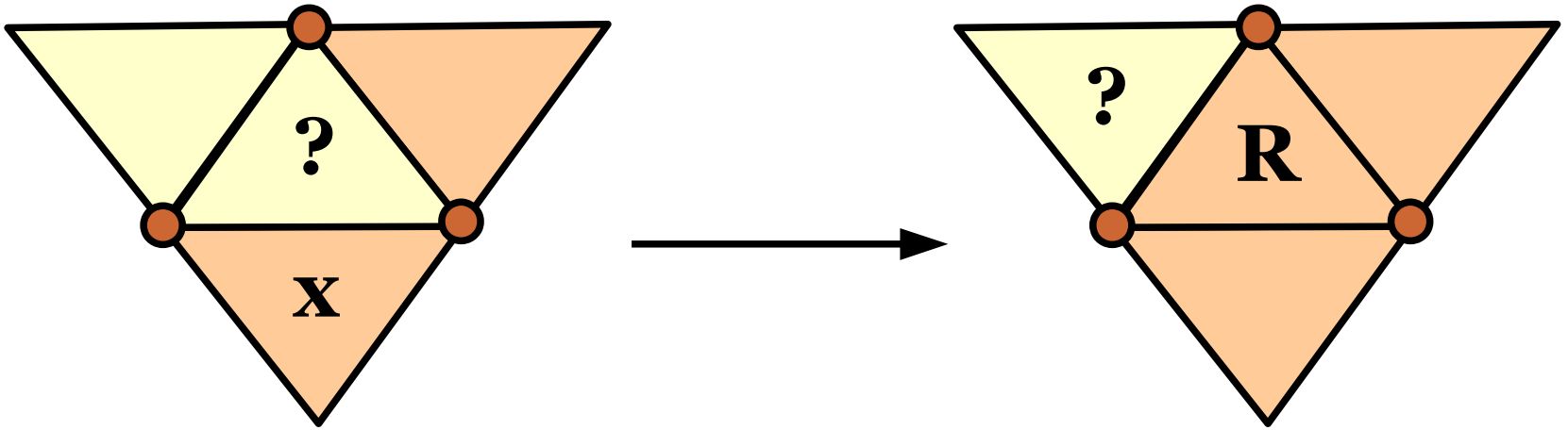

```
write( “L” );
c = c.r;
```

CLERS kódování – E



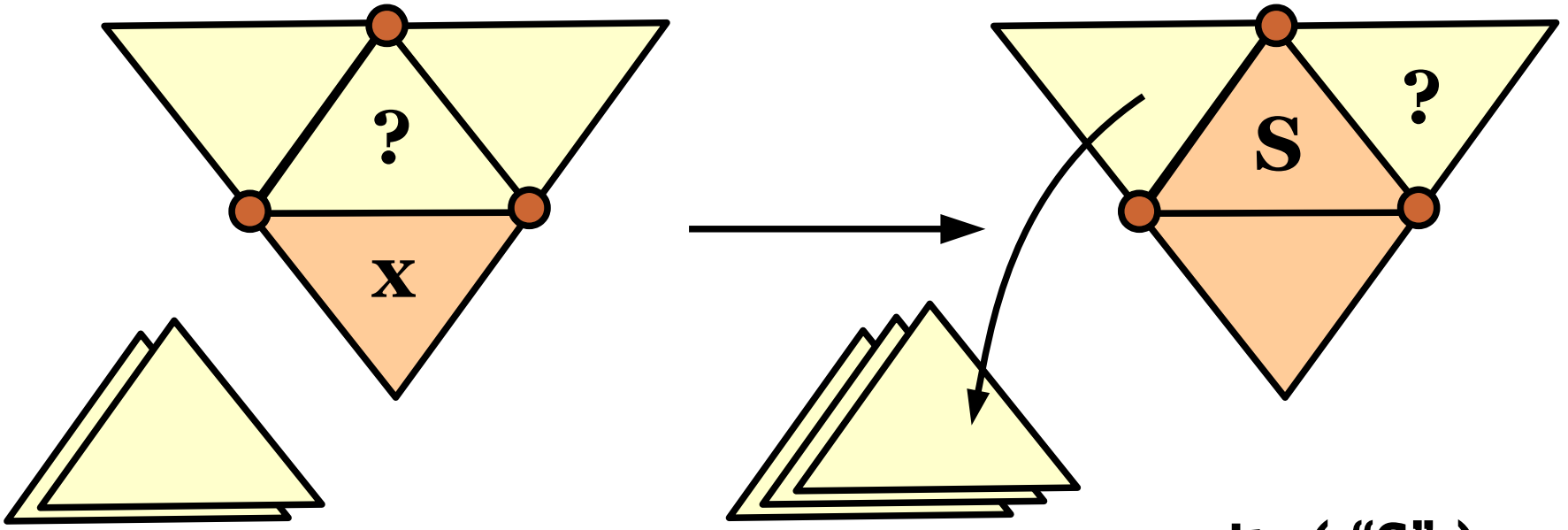


CLERS kódování – R




```
write( “R” );
c = c.l;
```

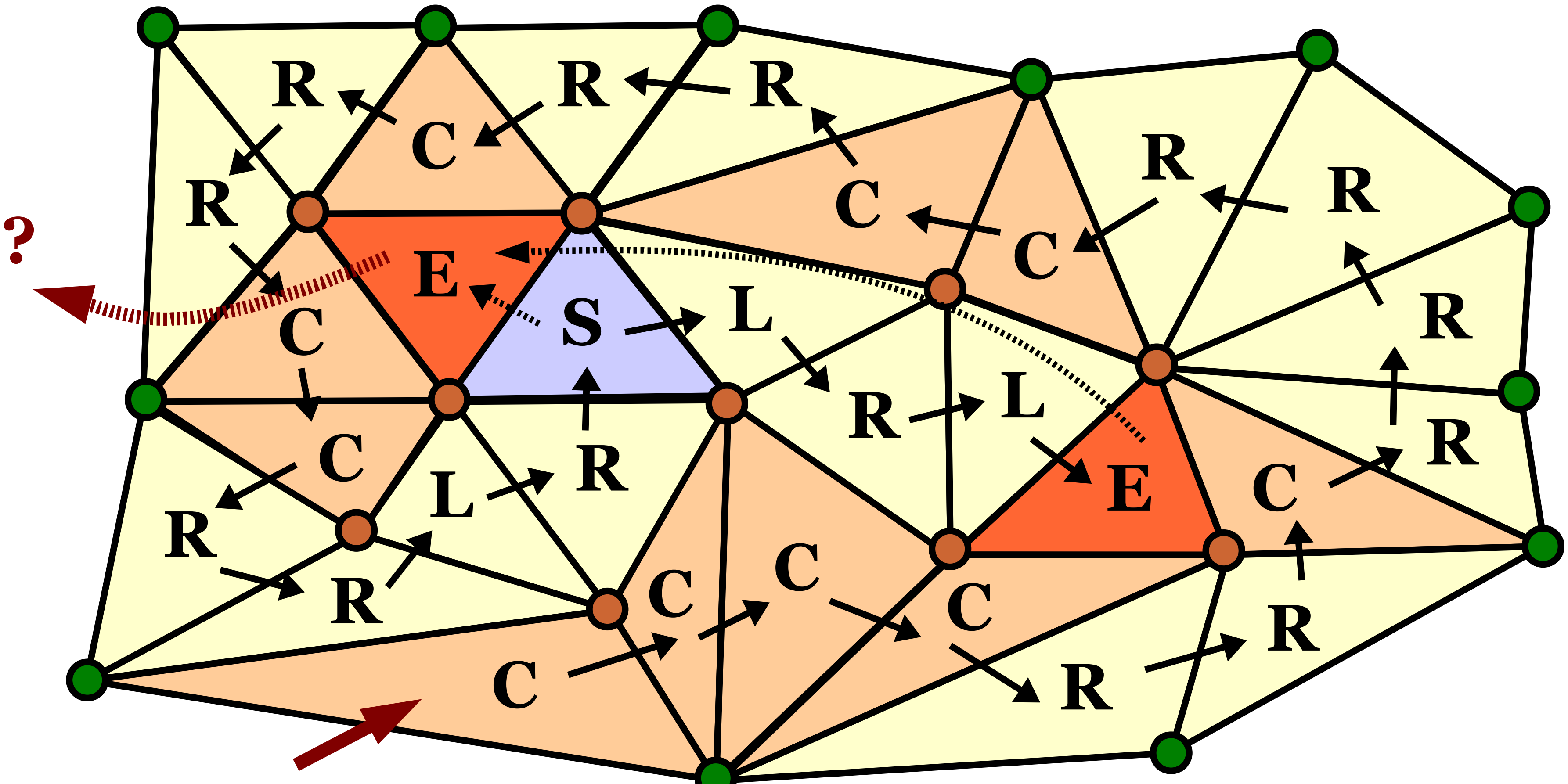
CLERS kódování – S




```
write( “S” );
push( c.l );
c = c.r;
```

Průchod sítí – příklad

(obvod oblasti již byl navštíven ●)



CCCCRRCRRRRCCRRCRRCCRRLRSLRLEE..

Kódování průchodu

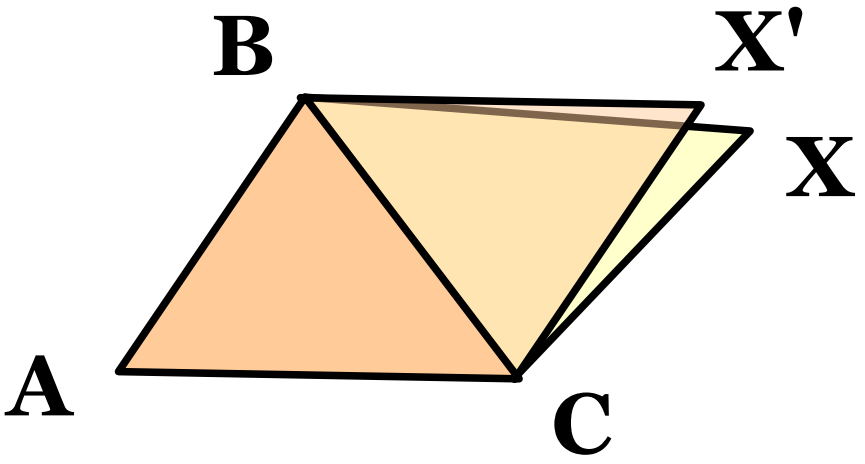
- jedině krok „**C**“ navštěvuje nový vrchol
  - tj. přibližně **½ kroků** musí být typu „**C**“
  - statický kód: **C=0**, **L=110**, **E=111**, **R=101**, **S=100**
  - amortizovaná délka kódu bude maximálně **2t**
  - příklad: 10·1 + 20·3 = 70 bitů na 30 trojúhelníků
- efektivnější statický kód (< 1.84t)
  - kroky následující za „**C**“: **C=0**, **S=10**, **R=11**
  - bez předcházejícího „**C**“: existují 3 kódy, z nichž vždy aspoň jeden dosahuje délky **< 11t/6**
  - jeden z nich: **C=00**, **L=110**, **E=01**, **R=10**, **S=111**

Kódování průchodu

- další metoda (≤ **2t**, průměrně mezi **1.3t** a **1.6t**)
  - dvojice kroků: **CR=01**, **CC=00**, **CS=1101**
  - jednotlivé kroky: **L=1110**, **E=1100**, **R=10**, **S=1111**
  - příklad: (00)(00)(10)(10)(01)(10)(10)(10)(00)(10)(10)(01)(10)(00)(10)(10)(1110)(10)(1111)(1110)(10)(1110)(1100)(1100) … 60 bitů na 30 trojúhelníků (10 vrcholů)
- dynamické entropické kódování CLERS řetězce
  - např. Huffmanův kód (včetně přenosu Huff. stromu) – redukce až na **< 1.0t**
  - dynamické kódování (aritmetický kód) může být ještě efektivnější

Kódování geometrie

- predikce z předchozího navštíveného vrcholu
  - nebo z obou již navštívených vrcholů téhož trojúhelníka
  - očekávaná úspora: ½ bitů pro reprezentaci polohy
- predikce ze sousedního trojúhelníka
  - rovnoběžník: $\mathbf{X} \approx \mathbf{X'} = \mathbf{B} + \mathbf{C} - \mathbf{A}$



Kódování geometrie

- predikce úhlu mezi dvěma sousedními trojúhelníky
  - může ještě dále vylepšit rovnoběžníkové pravidlo
  - predikce ze sousedních trojúhelníků nebo z globální statistiky sítě
- predikce z několika předchozích trojúhelníků
  - čtyři spojití předchůdci
  - lineární koeficienty predikce se optimalizují a přenášejí v hlavičce souboru
- v příznivých případech se dosahuje až **7 bpv** pro kódování geometrie

Literatura

- **http://www.gvu.gatech.edu/~jarek/papers.html**
  (Jarek Rossignac's publications)
- **http://www.gvu.gatech.edu/~jarek/edgebreaker/**
  (Jarek Rossignac - edgebreaker)