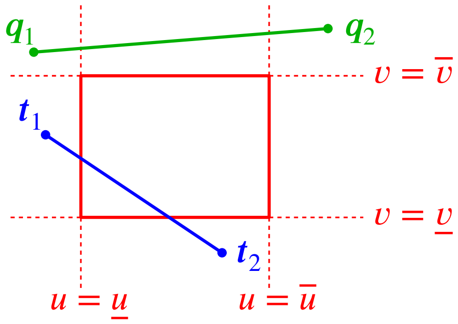
5.2.2 Strecken-Clipping nach Cohen und Sutherland

Idee: Manchmal ist sehr leicht feststellbar, dass die Strecke eine bestimmte Fensterseite gar nicht schneiden kann oder sogar völlig innerhalb bzw. außerhalb des Fensters liegt.



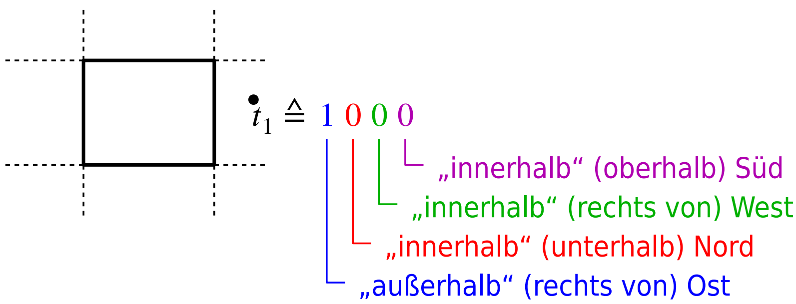
- $\overline{q_1 q_2}$ ist unsichtbar, denn beide Endpunkte liegen oberhalb der **Geraden** $v = \overline{v}$.
- $\overline{t_1 t_2}$ kann die Nordseite des Fensters nicht schneiden, da beide Endpunkte auf derselben Seite der Geraden $v = \overline{v}$ liegen.

  (analog: kein Schnitt mit „Ost“)

klar: Eine Strecke s liefert höchstens dann einen Schnittpunkt mit der Nord- (West-, ...) Seite des Fensters, wenn die Endpunkte von s auf verschiedenen Seiten der entsprechenden Geraden liegen.

Ziel: diese Tests möglich effizient durchführen

Ansatz: Ordne jedem Endpunkt der Strecke einen **4-stelligen Binärcode** zu, der die Lage des Punktes bzgl. der vier „Fenstergeraden“ angibt:



0 1 1 0	0 1 0 0	1 1 0 0
0 0 1 0	0 0 0 0	1 0 0 0
0 0 1 1	0 0 0 1	1 0 0 1

Algorithmus 5.5:

```
Algorithmus Cohen/Sutherland

wiederhole
   code₁ := (u₁ > ū, v₁ > v̄, u₁ < u̲, v₁ < v̲)
   code₂ := (u₂ > ū, v₂ > v̄, u₂ < u̲, v₂ < v̲)
   wenn (code₁ and code₂) ≠ 0000
      fertig: Strecke unsichtbar      // beide Endpunkte liegen „außerhalb“ derselben Geraden
   sonst
      wenn (code := (code₁ or code₂)) = 0000
         fertig: Strecke sichtbar        // beide Endpunkte liegen im Fenster
      sonst
         sei i das erste 1-Bit in code
         bestimme den Schnittpunkt q der Strecke q₁q₂ mit der Fenstergeraden zu Bit i
         wenn Bit i in code₁ gesetzt ist
            ersetze q₁ durch q
         sonst
            ersetze q₂ durch q
bis fertig
```


Berechnung der Schnittpunkte z. B. mit der Steigungsform

$$v = v_1 + \underbrace{\frac{v_2 - v_1}{u_2 - u_1}}_{=:\, m_v} \cdot (u - u_1)$$

bzw.

$$u = u_1 + \underbrace{\frac{u_2 - u_1}{v_2 - v_1}}_{=:\, m_u} \cdot (v - v_1)$$

durch Einsetzen von $u = \underline{u}$, usw.

⇒ 4 Additionen + 1 Multiplikation + 1 Division, wenn die Steigung berechnet werden muss
2 Additionen + 1 Multiplikation, sonst

(m_u und m_v werden höchstens einmal berechnet.)

⊕ sehr schnell, wenn die meisten Strecken bereits beim ersten Test akzeptiert oder abgelehnt werden können

⊖ relativ viele Operationen, wenn mehrere Schnittpunkte bestimmt werden müssen

⊖ nicht effizient, wenn nicht gegenüber einem achsenparallelen Rechteck geclippt wird