

Übung 8

Blatt 7, PL/1, Krypto

Übungsblatt 7

Prädikatenlogik

Krypto



- Newton $g = m * M * G / d^2$

- $f = M * a$

- $a = m * G / d^2$

- $(x1, y1, z1) \quad (x2, y2, z2)$

- Umsetzung

- Gravitational acceleration of the first body

- induced by the second body

```
gravityAccel :: Body -> Body -> [Double]
```

```
gravityAccel (Body a _ _) (Body b _ m)
```

```
  = let s = b `vDiff` a in
```

```
      let mag = constMyG * m / ((norm s)3) in
```

```
          mag `scale` s
```



```
gravityAccels body :: Body -> [Double]
```

```
-- Individual accelerations of the first body  
-- induced by the list of bodies
```

```
gravityAccels :: Body -> GravState -> [[Double]]
```

```
gravityAccels body bodies  
= map (gravityAccel body) bodies
```



```
-- Overall acceleration of the first body
-- induced by the list of bodies
totalAccel :: Body -> GravState -> [Double]
totalAccel body bodies
  = foldr vSum (0 0 0)
    (gravityAccels body bodies)
```



a

$$v_1 = v_0 + \delta_T * a$$

$$s_1 = s_0 + \delta_T * v + 0.5 \delta_T^2 * a$$

-- Compute new position according to gravity given

-- the time step, the body and list of other bodies

gravityUpdate :: Double -> Body -> GravState -> Body

gravityUpdate deltat (Body s v m) bodies

= let a = totalAccel (Body s v m) bodies in

Body (s `vSum` (deltat `scale` v) `vSum`

((deltat*deltat/2) `scale` a)

(v `vSum` (deltat `scale` a))

m



- [1,2,3,4,5] []
- [2,3,4,5] [1]
- [3,4,5] [2,1]
- ...
- [] [5,4,3,2,1]

-- Compute new positions according to gravity
-- given the time step,
-- the list of bodies still to update and
-- the list of bodies already processed

gravityUpdates

:: Double -> GravState -> GravState -> GravState

gravityUpdates deltat (x:xs) prev

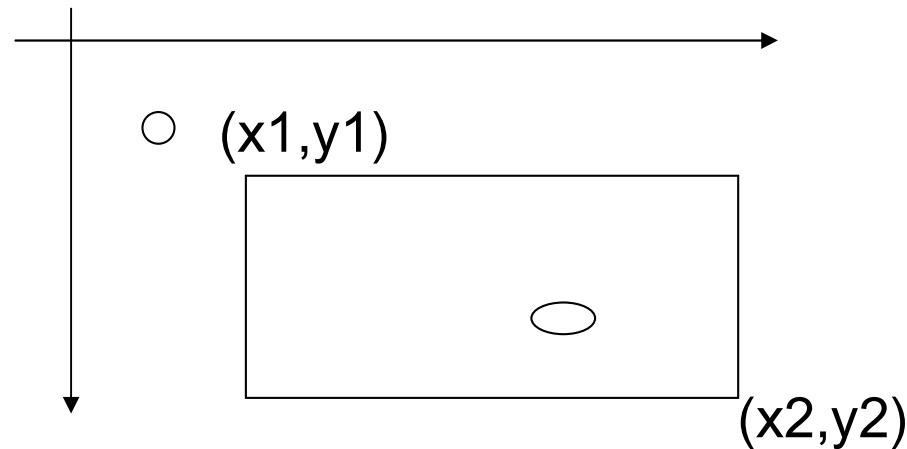
= gravityUpdate deltat x (prev ++ xs) :

gravityUpdates deltat xs (x:prev)

gravityUpdates deltat [] moment = []



- $(x - x1)/(x2-x1)$
- $(y - y1)/(y2-y2)$



```
-- Display planet, given a viewport,
-- a view specification and body data
planetView :: (Double,Double,Double,Double) ->
             (ViewSpec, Body) -> Graphic
planetView (xmin,ymin,xmax,ymax)
           ((color,radius), (Body s _ _ ))
= withColor color (
  circle (round(((s!!0)-xmin)*screenWidth / (xmax-xmin)),
          round(((s!!1)-ymin)*screenHeight / (ymax-ymin)))
          radius )
```



- [S1, S2, ... SN] [B1, B2, ... BN]
- [(S1,B1) , (S2, B2), ... (SN, BN)]

```
-- Display all planets in a given viewport
solarView :: (Double,Double,Double,Double) ->
            ViewSpecs -> GravState -> [Graphic]
solarView bounds atts xs
= map (planetView bounds) (zip atts xs)
```



- Schreibweise für Ersetzungen:
 $P [x/y]$ bedeutet: x für y einsetzen
(Eselsbrücke: y herauskürzen)
- Umbenennen gebundener Variabler
(Voraussetzung: y nicht frei in P)
 - $\forall x: P$ wird zu $\forall y: P [y/x]$
 - $\exists x: P$ wird zu $\exists y: P [y/x]$
- $\forall x : (x \vee \neg y) \quad [y/x]$
- $\forall y : (y \vee \neg y) = \text{True}$ **FALSCH SO NICHT!** Prämisse verletzt.



- Verschieben von Teilausdrücken
(Voraussetzung: x nicht frei in P / in P nicht an äußeres x gebunden)
 - $P \wedge \forall x: Q = \forall x: P \wedge Q$
 - $P \vee \forall x: Q = \forall x: P \vee Q$
 - $P \wedge \exists x: Q = \exists x: P \wedge Q$
 - $P \vee \exists x: Q = \exists x: P \vee Q$

- $\neg x \wedge \forall x: x \rightarrow \forall x: x \wedge \neg x = \text{True}$ **FALSCH SO NICHT!**

- $\forall x: y \vee x \rightarrow y \vee \forall x: x$



- Negation:

- $\neg \forall x: P = \exists x: \neg P$

- $\neg \exists x: P = \forall x: \neg P$

(Eselsbrücke: Quantor negieren, wenn ein Nicht ihn überquert)

- $\neg \forall a \exists l \forall y \exists k: P$

- $\exists a: \neg \exists l \forall y \exists k: P$

- ...

- $\exists a \forall l \exists y \forall k: \neg P$



▪ Vertauschen von Quantoren:

- $\forall x \forall y : P = \forall y \forall x : P$
- $\exists x \exists y : P = \exists y \exists x : P$

- $\exists x \forall y : P \rightarrow \forall y \exists x : P$
Aber nicht die Gegenrichtung!

- Warum?
 $\forall y \exists x : P \neq \exists x \forall y P$

$X \setminus Y$	0	1
0	1	
1		1

Linke Seite:

jede Spalte enthält eine 1

Rechte Seite:

eine Zeile enthält nur 1en



- Aggregation von \forall
 - $(\forall x : F) \wedge (\forall x : G) = \forall x : (F \wedge G)$
 - $(\forall x : F) \vee (\forall x : G) = \forall x : (F \vee G)$

- Keine Aggregation von \exists !

- Gemischte Aggregation:
 - $(\forall x : F) \wedge (\exists x : G) \rightarrow \exists x : (F \wedge G)$
 - $(\forall x : F) \vee (\exists x : G) \rightarrow \exists x : (F \vee G)$
 - Nur diese Fälle.
 - Rückrichtung gilt nicht!



- Eine Formel F ist in Pränexnormalform gdw.
 $F = Q_1 Q_2 \dots Q_n : G$ mit $Q_i \in \{\forall, \exists\}$ und G quantorenfrei
- Jede Formel in PL/1 kann in Pränexnormalform überführt werden.
- Bereinigung:
Alle aussagenlogischen Operatoren in \wedge, \vee, \neg überführen
- Beispiel:
 - $\neg \forall x : (P(x) \wedge \exists y : (Q(x,y) \vee \forall z : R(y,z)))$
 - Lösung siehe nächste Seite.



- $F = P S$
 - Präfix P enthält nur Allquantoren
 - Suffix S ist quantorenfrei
- Existiert für alle Formeln in PL/1. Wie einführen?
 - Start mit Pränexform: $\forall x \forall y \dots \forall z \exists w$
 - Mit Auswahllemma: $w = sk_w(x, y, \dots, z)$
 - Quantor streichen, Variable durch Funktion ersetzen (Skolemfunktion)

- Beispiel: $\forall y \exists x \exists z : \neg (P(x) \wedge (Q(x,y) \vee R(y,z)))$
 - $\forall x \forall y \exists z : z = x - y \wedge Z(x, z, y)$
 - $sk_z = x - y$
 - $\forall x \forall y Z(x, sk_z(x, y), y)$



- Kunst der Verschlüsselung
- Nachricht
- Plaintext = unverschlüsselte Nachricht
- Ciphertext = verschlüsselte Nachricht



- Substitution
 - Zeichenweise Ersetzung $f :: a \rightarrow a$

 - Unterformen
 - Monoalphabetisch (ein und dieselbe Ersetzung für alle Zeichen eines Strings)
 - Caesar
 - Polyalphabetisch

- Beispiel: Caesar mit -3 angewandt auf FDHVDU
- CAESAR



- Transposition
 - Verwürfelung
 - feste Blocklänge n
 - Auffüllen zulässig
 - Permutation der Zahlen $0 \dots n-1$

- Beispiel: $[2,3,1,0]$ angewandt auf
- 0123 0123
- DCAB HGEF (Ciphertext)
- ABCD EFGH (Plaintext)

- IXXX
- DCAB HGEF XXIX



- Bauer, Kryptologie - Methoden und Maximen, Springer.
 - Gute Behandlung aus mathematischer Sicht
- Schneier, Applied Cryptography, Wiley.
 - Schwerpunkt auf praktische Umsetzung
- Kahn, The Codebreakers, Scribner.
 - Packende Historie

