



# Hoorcollege III: moleculaire dynamica

Inge Bethke

CSP

Faculteit NWI

Instituut voor Informatica

Universiteit van Amsterdam

<http://www.science.uva.nl/~inge/>



## Moleculaire Dynamica

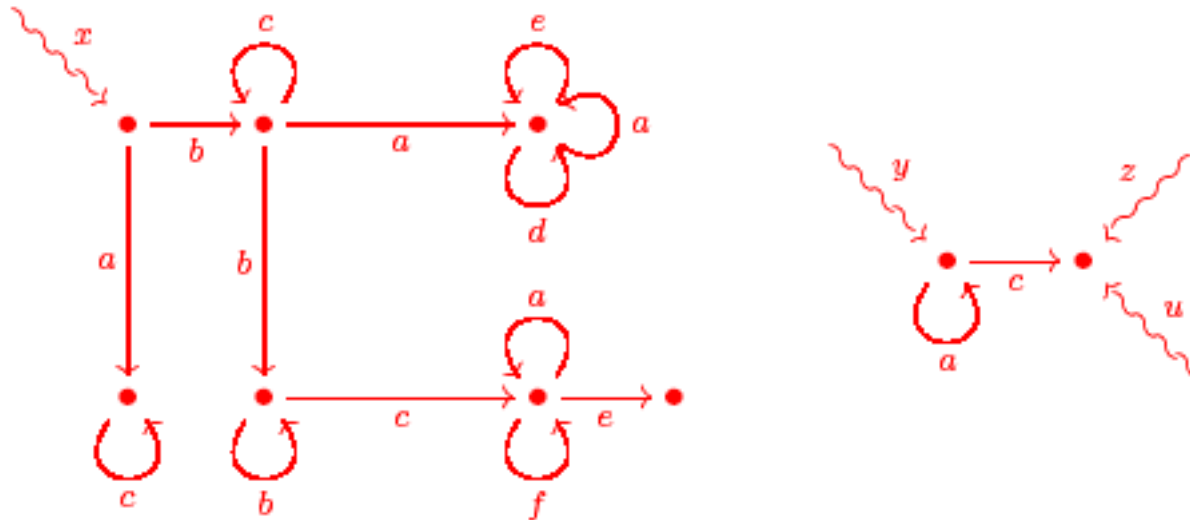
- Voorbeeld van een concrete invulling van de impliciete parameter  $A$ , de verzameling van basisinstructies.
- Doel: het modelleren van toestanden en toestandsveranderingen van computerprogramma's.

## Atomen, moleculen en vloeistoffen

- Toestand (van een computerprogramma): vloeistof bestaande uit een aantal moleculen, d.w.z.
  - ★ *atomen*,
  - ★ *velden*.
- Toestandsverandering: het aangaan van nieuwe of het verbreken van bestaande bindingen, d.w.z.
  - ★ toevoegen of verwijderen van velden,
  - ★ veranderen van het doel van velden.
- Observatie van toestanden en veranderingen: *focus*.

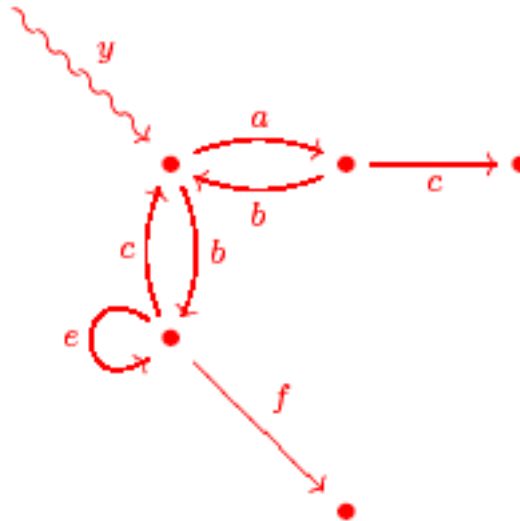
## Atomen, moleculen en vloeistoffen

Eenvoudige diagrammen: vloeistof bestaande uit 2 moleculen



## Atomen, moleculen en vloeistoffen

### Molecule met dubbele bindingen



## Instructies en acties

**MPP** (molecular programming primitives): 4 mutaties, 2 toekenningen en 2 tests.

*Mutaties.*

*Atoomcreatie*  $x = new$



*Veldintroductie*  $x.+f$



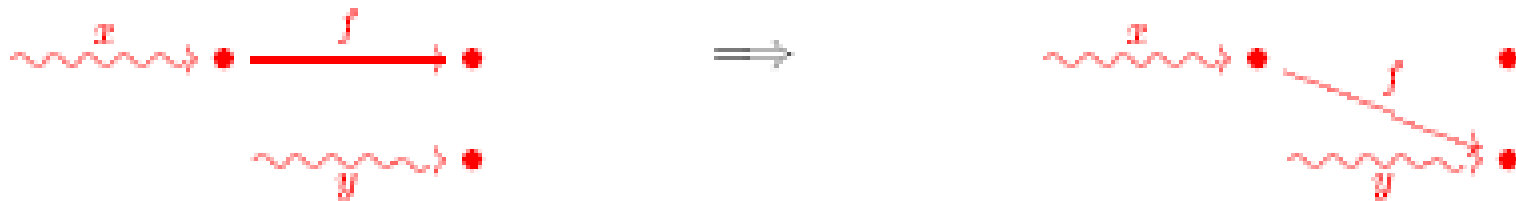
## Instructies en acties

### *Mutaties.*

*Veldverwijdering  $x.-f$*



*Veldmutatie  $x.f = y$*



### Instructies en acties

*Toekenningen.*

*Veldselectie  $x = y.f$*



*Toekenning  $x = y$*

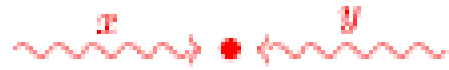




## Instructies en acties

*Tests.*

*Atoomidentiteitstest*  $x == y$



true

*Veldtest*  $x/f$



true

## Instructies en acties

Aaneenschakeling van instructies:

$$X = x = \text{new}; y = \text{new}; x.+a; y.+b; x.a = y; y.b = x$$

*(demo1)*

$$X = x = \text{new}; \text{\textcircled{0}}; y = \text{new}; y.+a; y.a = x; x = y; \#\#\text{\textcircled{0}}$$

*(demo2)*

**Rekenen zonder getallen (demo3 + demo4)**

Model 1: *setZero* creëert de 0, *pred* berekent de voorganger en *succ* de opvolger.

Afspraak: de argumenten van unaire (binaire) functies hebben de focusnaam *x* (*x1* en *x2*), het resultaat heet *y*.

$$\textit{setZero} = Z = \textit{new}; y = Z$$

$$\textit{pred} = -x/p\{; y = x; \}\{; y = x.p; \}$$

$$\begin{aligned} \textit{succ} = & -x/s\{; x.+s; y = \textit{new}; y.+p; y.p = x; x.s = y; \\ & \}\{; y = x.s; \} \end{aligned}$$

**Rekenen zonder getallen (demo5)**

Optelling in model 1: we kiezen voor een implementatie die correspondeert met de recursieve definitie

$$x1 + x2 = \begin{cases} x1 & \text{als } x2 = 0, \\ (x1 + 1) + (x2 - 1) & \text{anders.} \end{cases}$$

*add* =  $\mathcal{L}0; +x2 == Z\{; y = x1; \}\{; x = x1; succ; x1 = y;$

$x = x2; pred; x2 = y; ## \mathcal{L}0; \}$

**Rekenen zonder getallen (demo6)**

“modulus 2” in model 1: we kiezen voor een implementatie die correspondeert met de recursieve definitie

$$x \bmod 2 = \begin{cases} 0 & \text{als } x = 0, \\ (x - 2) \bmod 2 & \text{als } x - 1 > 0, \\ x & \text{anders.} \end{cases}$$

*mod2* =  $y = x; \mathcal{L}1; -y == Z\{; x = y; pred;$

$-y == Z\{; x = y; pred; \#\#\mathcal{L}1; \}\{; y = x; \}; \}\{; \}$

**Rekenen zonder getallen (demo7 + demo8 + demo9)**

Model 2: uitbreiding van model 1 met *even*- en *odd*-velden

*setZero* =  $Z = \text{new}; Z.+even; y = Z$

*pred* =  $-x/p\{;y = x;\}\{;y = x.p;\}$

*succ* =  $-x/s\{;x.+s; y = \text{new}; y.+p; y.p = x; x.s = y;$

$+x/even\{;y.+odd;\}\{;y.+even;\};\}\{;y = x.s;\}$



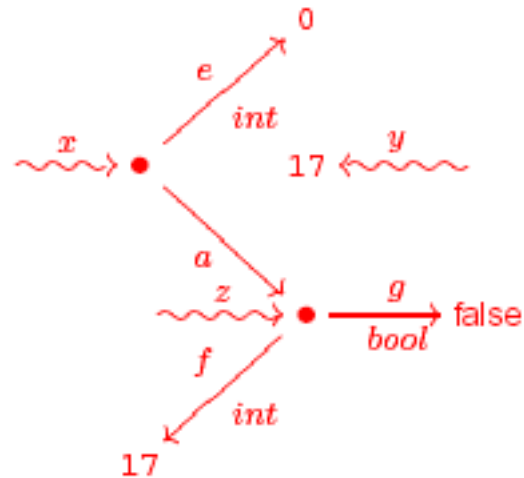
## Waarden

Waarden:

- vast en eindig aantal *basistypen* : *bool*, *int*, ... ,
- *literalen* (benaming van waarden): *true*, *false*, ... , *0*, *1*, ... ,
- waarden worden beschouwd als terminale objecten, d.w.z. objecten zonder uitgaande verbindingen.

## Waarden

Een molecule met waardevelden:





## Waarden

**MPPV** is de uitbreiding van **MPP** met 4 mutaties

*Waardeveldintroductie* :  $x.+f : t.$

*Waardeveldverwijdering* :  $x.-f : t.$

*Waardeveldmutatie* :  $x.f = y.$

*Waardeveldmutatie met constante* :  $x.f = u.$



## Waarden

en 3 toekenningen en 3 tests

*Waardeveldselectie* :  $x = y.f$ .

*Waardetoekenning* :  $x = y$ .

*Waardetoekenning met constante* :  $x = u$ .

*Waarde-identiteitstest* :  $x == y$ .

*Waarde-identiteitstest met constante* :  $x == u$ .

*Waardeveldtest* :  $x/f : t$ .



## Waarden (demo10)

Voorbeeld:

$$X = x = new; z = new; y = 17; x.+e : int; x.+a; z.+g : bool;$$
$$x.e = 0; x.a = z; z.g = true; z.+f : int; z.f = y$$

**Rekenen met waarden (demo 11 + demo12 + demo13)**Basisprogramma's *NEG*, *AND* en  $I_{b_0, \dots, b_n}$ :

$$I_{b_0, \dots, b_n} = i = new; i.+w_0 : bool; i.w_0 = b_0; \dots; i.+w_n : bool; i.w_n = b_n$$

$$\begin{aligned} NEG &= neg = new; neg.+i; neg.i = i; o = new; o.+w_0 : bool; \\ & i_0 = i.w_0; +i_0 == true\{\}; o.w_0 = false;\}\{\}; o.w_0 = true;\}; \\ & neg.+o; neg.o = o \end{aligned}$$

$$\begin{aligned} AND &= and = new; and.+i; and.i = i; o = new; o.+w_0 : bool; \\ & i_0 = i.w_0; i_1 = i.w_1; \\ & +i_0 == true\{\}; +i_1 == true\{\}; o.w_0 = true;\}\{\}; o.w_0 = false;\}; \\ & \}\{\}; \\ & and.+o; and.o = o \end{aligned}$$