

# Beweren en bewijzen/2010-11/werkstuk/Windkorenmolen De Vlijt Wageningen

---

Project state::busy

Titel::Windkorenmolen "De Vlijt", Wageningen

[http://upload.wikimedia.org/wikipedia/commons/thumb/a/a7/Molen\\_Wageningen\\_de\\_Vlijt\\_Windmill.jpg/264px-Molen\\_Wageningen\\_de\\_Vlijt\\_Windmill.jpg](http://upload.wikimedia.org/wikipedia/commons/thumb/a/a7/Molen_Wageningen_de_Vlijt_Windmill.jpg/264px-Molen_Wageningen_de_Vlijt_Windmill.jpg)

Course::Beweren en bewijzen Academisch jaar::2010-11

,

Mijn werkstuk gaat over windmolens. Hoewel er nog weinig werkende windmolens zijn in Nederland, is er wel eentje in de stad waar ik ben opgegroeid, een stellingmolen, en ik heb er een paar jaar gewerkt toen ik op de middelbare school zat. In het specifiek ga ik het in dit werkstuk hebben over deze molen, Windkorenmolen "De Vlijt", hoewel het gezien mijn focus niet echt veel verschil maakt of ik nu een andere windmolen had gekozen of niet.

## Focus

### Fragment v/d realiteit

Ik beschouw enkel de meest essentiële onderdelen van de molen die te maken hebben met het maken van meel, van het begin tot het eind van het proces: de elevator die het graan omhoog brengt, de molenstenen die het graan maalt, de molenaar die de zakken met meel vervangt ter voorkoming van overstroming van die zakken. En natuurlijk abstraheer ik vervolgens ook erg van zaken die er niet toe doen zoals de vorm van de elevator, de haarkleur van de molenaar, ed..

### Gezichtspunt

Ik neem aan dat de molenaar inmiddels zijn windmolen heeft staan, goed ontbeten heeft, de wind blaast, de molen werkt en houdt me vervolgens bezig met het proces van graan malen. De molenaar wil tenslotte van zijn molen zeker weten dat er een bepaalde hoeveelheid zakken zullen zijn gevuld, na een bepaalde tijdsduur, gegeven dat hij aan het begin van de dag een zekere hoeveelheid graan heeft.

En dit is precies wat ik heb gedaan: een verificatiemodel opgesteld zodat de molenaar dit zeker kan weten.



## Inleiding

*De inleiding en conclusie schrijf je als laatste, niet eer je weet wat in te leiden is.*

### Waarom?

De molen is een prachtig werk van vernuft en natuurlijk ideaal als te onderzoeken artefact in dit vak. Er is een beetje domeinkennis nodig, maar dat is geen probleem: ik heb een paar jaar in een windmolen in Wageningen gewerkt en kan daar makkelijk aankloppen voor informatie; bovendien woon ik hier in Nijmegen naast een windmolen, waar ik ook kan binnenlopen voor informatie.

### Wat wordt formeel behandeld?

De tijd, hoeveelheid graan, stroomtoevoer en hoeveelheid gevulde zakken meel. Waarbij de tijd en de hoeveelheden van essentieel belang zijn en de stroomtoevoer een stuk minder.

Wat ik bijvoorbeeld niet formeel ga behandelen in dat de wieken werken, de wind goed staat, etc.

### Vereenvoudigingen

- [..Secondes, gram graan/meel? -- of reeele getallen?...]
- Gebeurtenissen vinden heel exact op moment plaats, ik abstraheer van speling.
- Ik abstraheer van kleine beetjes graan en meel die verloren gaan.
- [..TODO..]

## Structuur

### Onderdelen

naam	korte informele specificatie
Elevator	Verplaatst graan naar de schuddebak.
Schuddebak	Dit onderdeel behelst eigenlijk de combinatie van een kleine silo of opslagplaats met de schuddebak, zorgt voor een buffer als het ware en regelt vervolgens dat graan gelijkmatig over de tijd naar de molenstenen gaat.
Molenstenen	Maalt inkomend graan tot meel
Meelpijp	Leidt het meel naar de zak.
Molenaar	Zodra de zak bijna helemaal vol is vervangt de molenaar deze zak met een nieuwe, lege zak en plaatst de volle zak vervolgens bij alle andere gevulde meelzakken.

### Het domeinmodel

Types			
naam		betekenis	
T		Tijd: een verzameling van alle positieve (inclusief 0) tijdstippen in secondes, vanaf start van het werk	
G		Graan: een verzameling van alle positieve (inclusief 0) hoeveelheden graan in gram	
M		Meel: een verzameling van alle positieve (inclusief 0) hoeveelheden meel in gram	
ZAKKEN		Zakken: een verzameling van alle positieve (inclusief 0) aantallen met meel gevulde zakken	
Constanten			
naam	type	betekenis	waarde
eG	G	De hoeveelheid graan die in een back past	
eD	T	De tijd die het duurt tot de volgende bak langskomt	
eTD	T	De totale tijd die het duurt tot een bak boven is	
pCap	M	De capaciteit, dus de maximale hoeveelheid meel, dat per seconde door de meelpijp kan	
pTD	T	De tijdsduur die het duurt tot meel van de ingang naar de uitgang van de meelpijp gaat	
Proposities			
naam		betekenis	meting
stroom		Er gaat stroom naar de elevator	Volt- / Stroommeter
Predicaten			
naam	type	betekenis	meting
graan	$T \rightarrow G \rightarrow B$	$\text{graan } t \ g \implies$ Er is op tijdstip $t$ $g$ graan onderaan de lift	Wegen
kaap	$T \rightarrow G \rightarrow B$	$\text{kaap } t \ g \implies$ Er is op tijdstip $t$ $g$ graan in de kaap (bovenaan de lift)	Wegen
geschud	$T \rightarrow G \rightarrow B$	$\text{geschud } t \ g \implies$ Er is op tijdstip $t$ $g$ graan door de schuddebak naar de molenstenen gestuurd	Wegen (voorzichtig!)
gemalen	$T \rightarrow M \rightarrow B$	$\text{gemalen } t \ m \implies$ Er is op tijdstip $t$ $m$ meel gemalen en bij de invoer van de meelpijp	Wegen
naarzak	$T \rightarrow M \rightarrow B$	$\text{naarzak } t \ m \implies$ Er is op tijdstip $t$ $m$ meel bij de uitvoer van de meelpijp	Wegen
inzak	$T \rightarrow M \rightarrow B$	$\text{inzak } t \ m \implies$ Er is op tijdstip $t$ $m$ meel in de zak	Wegen

## Hulpdefinities

Optioneel. Samengestelde predicaten, **gedefinieerd op basis van de predicaten uit het domeinmodel**.

## Onderdeelspecificaties

### Elevator

Definition Elevator :=

```

stroom -> (
  forall t:T, (exists k:Z, k >= 0 /\ t = k * eD) ->
  (
    (* Elke {eD} sec wordt {eG} gram graan meegenomen door de elevator *)
    (forall g:G, (g > eG -> graan t g -> graan (t + eD) (g - eG))
  )
)
```

```

        /\ (g <= eG -> graan t g -> graan (t + eD) 0)

    )

    (* Elke {eTD} sec later arriveert deze bovenaan *)
    /\ (forall g:G, g > eG -> (forall gk:G, graan t g /\ kaap (t + eTD - eD) gk -> kaap (t + eTD) (gk + eG))
        /\ g <= eG -> (forall gk:G, graan t g /\ kaap (t + eTD - eD) gk -> kaap (t + eTD) (gk + g))
    )

    (* Alle tijdstippen tussen n*eD en (n+1)*eD *)
    /\ (forall g:G, forall s:T, (s >= t /\ s < t+eD) ->
        (graan s g <-> graan t g) /\ (kaap s g <-> kaap t g))
))
.

```

### Uitleg

Eerst moet worden opgemerkt dat de formule omhult wordt door een conditie, namelijk dat er stroom is, dit is vrij logisch. Wat als de stroom uitvalt halverwege het proces? Dit modelleer ik niet, want het is niet mijn focus.

Vervolgens bestaat de formule uit een zetal beweringen, opgesplitst in drie groepen, die allemaal moeten gelden voor elk opeenvolgend bakje, dus voor elk tijdstip  $n \cdot eD$ .

Groep 1 houdt zich bezig met wat onderaan de elevator gebeurt: Elke keer als een nieuwe bak komt, wordt de hoeveelheid meel onderaan de elevator minder.

Groep 2 houdt zich bezig met wat bovenaan de elevator gebeurt: Elke keer als onderaan een bak "gevuld" wordt, zoals in groep 1 beschreven wordt, wordt precies dezelfde hoeveelheid die daar verdwijnt  $eTD$  seconden later bovenaan "gedeponeerd".

Groep 3 houdt zich bezig met alle tijdstippen die niet aan bod komen: de tijdstippen tussen de beschreven, kritische, tijdstippen. Het legt uit dat de hoeveelheden graan onderaan, en bovenaan, niet veranderen totdat de volgende bak komt.

### Meelpijp

```

Definition Meelpijp :=
  (gemalen 0 0) /\
  (naarzak 0 0) /\
  (forall t:T, forall m:M,
    (m > pCap -> gemalen t m -> naarzak (t + pTD) pCap)
    /\ (m < pCap -> gemalen t m -> naarzak (t + pTD) m)
  ).

```

### Uitleg

TODO

### Molenaar

```

Definition Molenaar :=
  (* Begintoestand: nog geen meel in zak, nog geen gevulde zakken *)
  (inzak 0 0) /\
  (zakken 0 0) /\
  (* De zak mag nooit overstromen *)

```

```

~(exists t:T, exists m:M, inzak t m /\ m >= zMax) /\
(* Verder geldt dat.. *)
(forall t:T, forall mprev:M, forall madd:M, forall zprev:ZAKKEN,
  (inzak t mprev /\ naarzak t madd /\ zakken t zprev)
  -> (
    (* Zak wordt enkel bijgevuld door toevoer stroom meel *)
    (
      inzak (t + 1) (mprev + madd) /\ zakken (t + 1) zprev) /\
    (* OF, indien al meer dan {zMin} meel in zak, nieuwe zak! *)
    (mprev >= zMin /\ inzak (t + 1) madd /\ zakken (t + 1) (zprev + 1))
  )
  ).

```

### Uitleg

De molenaar vervangt de zakken steeds, zodra die gevuld zijn.

De pijp zorgt voor een toevoerstroom van meel aan de zak, zo dat op elk tijdstip  $t$  (als natuurlijke getallen) een nieuwe hoeveelheid meel op plek ME is (bijvoorbeeld 7gram, 10gram, 3gram, 7gram, etc..).

Er wordt nu gedefinieerd voor elk tijdstip, hoe ME en ZF er het volgende tijdstip uitzien. Dit is altijd een van twee gevallen: OFWEL de zak wordt gewoon bijgevuld, dus MZ opgehoogd, en ZF blijft dan hetzelfde, OFWEL de zak is net vervangen tussen deze twee tijdstippen zodat MZ de nieuwe minimale inhoud heeft van {madd}, en er een gevulde zak bij is (ZF++).

### per onderdeel

de naam specificatie als formule, leesbaar opgemaakt verwoording van dezelfde specificatie in natuurlijke taal eventueel extra uitleg, keuzes, afwegingen korte verantwoording: hoe kun je nagaan of deze specificatie in het echt juist is ("verificatie")?

Voorbeeld:

Definition BEL :=  $(\forall t: T. \text{stroom}(t)) \Rightarrow (\forall t: T. \text{druk}(t, \text{knop1}) \Leftrightarrow \text{rinkel}(t))$ . De bel rinkelt precies zolang knop1 gedrukt is. Alles onder de voorwaarde dat er de hele tijd stroom is.

afweging

Er zijn ook bellen die alleen stroom nodig hebben zolang de knop gedrukt is. De "kleine a" is hier bewust breder gehouden om geen bellen uit te sluiten waar een computer in zit.

verificatie

Gebruiksaanwijzing van de bel.

## De specificatie van het geheel

specificatie als formule, leesbaar opgemaakt verwoording van dezelfde specificatie in natuurlijke taal

(Dit moet natuurlijk aansluiten bij de eerste specificatie, vooraan)

## Het bewijs

Require Import BenB.

(\* Types > Verzamelingen \*)

Definition T := Z. (\* Tijdstippen na start werk \*)

Definition G := Z. (\* Hoeveelheid graan in gram \*)

```

Definition M := Z. (* Hoeveelheid meel in gram *)

Definition ZAKKEN := Z. (* Aantal gevulde zakken *)

(* Ik wil met natuurlijke getallen werken, maar Coq geeft dan veel problemen,
   dus los ik het natuurlijk gewoon zo op: *)

Open Scope Z_scope.

Axiom Natuurlijk: (forall x:T, x >= 0) /\ (forall x:G, x >= 0)

                  /\ (forall x:M, x >= 0) /\ (forall x:ZAKKEN, x >= 0).

(* Of er (genoeg, etc) stroom naar de elevator gaat *)

Variable stroom: B.

Variable TODO: B.

Variables graan kaap geschud:    T -> G -> B.
Variables gemalen naarzak inzak: T -> M -> B.
Variable  zakken:                T -> Z -> B.

(* Constanten > Algemeen *)

Definition maxSec := 60 * 60 * 24. (* Aantal seconden in een dag *)

(* Constanten > Elevator *)

Definition eD:T := 3%Z.  (* tijd tot volgende bak *)
Definition eG:G := 500%Z. (* hoeveelheid graan per bak *)
Definition eTD:G := 30%Z. (* tijd tot bak boven is *)

(* Constanten > Meelpijp *)

Definition pCap:M := 200. (* De capaciteit, dus de maximale hoeveelheid meel, dat per seconde door de meelpijp kan *)
Definition pTD:T := 5.   (* De tijdsduur die het duurt tot meel van de ingang naar de uitgang van de meelpijp gaat *)

(* Constanten > Molenaar *)

Definition zMin:M := 15000. (* Minimale meel in zak (inclusief) *)
Definition zMax:M := 25000. (* Maximale meel in zak (exclusief) *)

Definition Elevator :=

  stroom -> (

    forall t:T, (exists k:Z, k >= 0 /\ t = k * eD) ->

      (

        (* Elke {eD} sec wordt {eG} gram graan meegenomen door de elevator *)

        (forall g:G, (g > eG -> graan t g -> graan (t + eD) (g - eG))

          /\ (g <= eG -> graan t g -> graan (t + eD) 0)

        )

        (* Elke {eTD} sec later arriveert deze bovenaan *)

        /\ (forall g:G, g > eG -> (forall gk:G, graan t g /\ kaap (t + eTD - eD) gk -> kaap (t + eTD) (gk + eG))

          /\ g <= eG -> (forall gk:G, graan t g /\ kaap (t + eTD - eD) gk -> kaap (t + eTD) (gk + g))

        )

      )
  )

```

```

    (* Alle tijdstippen tussen n*eD en (n+1)*eD *)

    /\ (forall g:G, forall s:T, (s >= t /\ s < t+eD) ->
        (graan s g <-> graan t g) /\ (kaap s g <-> kaap t g))

))

.

(*
    IN:  rotatie, GB
    UIT: GC
*)

Definition Schuddebak := TODO.

(*
    IN:  rotatie, GC
    UIT: MD
*)

Definition Molenstenen := TODO.

Definition Meelpijp :=
    (gemalen 0 0) /\
    (naarzak 0 0) /\
    (forall t:T, forall m:M,
        (m > pCap -> gemalen t m -> naarzak (t + pTD) pCap)
        /\ (m < pCap -> gemalen t m -> naarzak (t + pTD) m)
    ).

Definition Molenaar :=
    (* Begintoestand: nog geen meel in zak, nog geen gevulde zakken *)
    (inzak 0 0) /\
    (zakken 0 0) /\
    (* De zak mag nooit overstromen *)
    ~(exists t:T, exists m:M, inzak t m /\ m >= zMax) /\
    (* Verder geldt dat.. *)
    (forall t:T, forall mprev:M, forall madd:M, forall zprev:ZAKKEN,
        (inzak t mprev /\ naarzak t madd /\ zakken t zprev)
        -> (
            (* Zak wordt enkel bijgevuuld door toevoer stroom meel *)
            (
                inzak (t + 1) (mprev + madd) /\ zakken (t + 1) zprev) /\
            (* OF, indien al meer dan {zMin} meel in zak, nieuwe zak! *)
            (mprev >= zMin /\ inzak (t + 1) madd /\ zakken (t + 1) (zprev + 1))
        )
    ).

(*
    Specificatie van het geheel
    IN:  stroom, rotatie, GA
    UIT: ZF

```

\*)

Definition Geheel := TODO.



# Paginabronnen en auteurs

**Beweren en bewijzen/2010-11/werkstuk/Windkorenmolen De Vlijt Wageningen** *Bron:* <https://lab.cs.ru.nl/algemeen/index.php?oldid=157236> *Auteurs:* Hanno Wupper, Kelley van Evert

# Afbeeldingsbronnen, licenties en bijdragers

**Bestand:MolenDeVlijtWageningen.jpg** *Bron:* <https://lab.cs.ru.nl/algemeen/index.php?title=Bestand:MolenDeVlijtWageningen.jpg> *Licentie:* onbekend *Auteurs:* Kelley van Evert

# Licentie

---

voorwaarden van het opleidingsinstituut  
Copyright