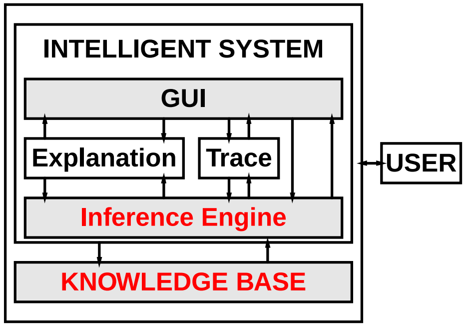
Intelligente Systemen & Logica

Intelligent systeem als kennissysteem:

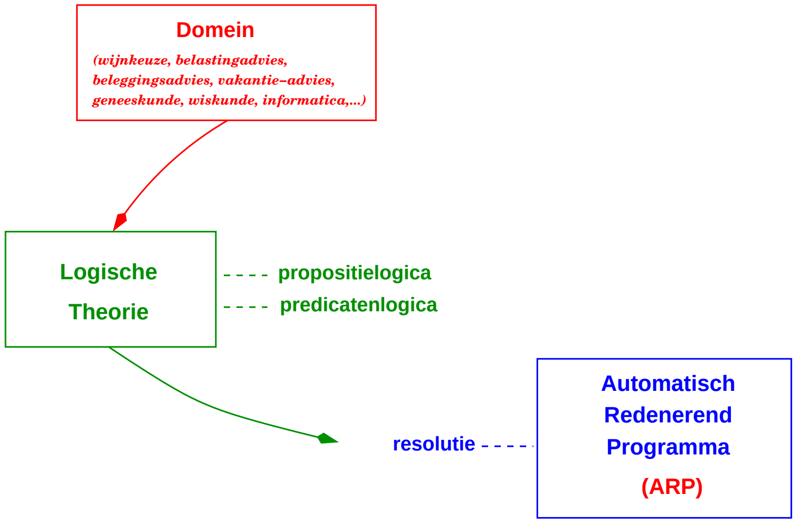
- kennisrepresentatie
- automatisch redeneren/inferentie
- acquisitie van kennis
- modelleren
- communicatie (systeem-gebruikersdialoog)
- ...

Architectuur

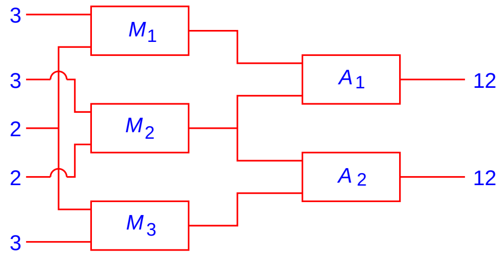


- **Kennisrepresentatie:**
  - Logica: $\forall x(C(x) \rightarrow E(x))$
  - Kansrekening: $P(\mathrm{E} \mid \mathrm{C}_1, \ldots, \mathrm{C}_n)$
- **Redeneren/inferentie:**
  - Deductie: $\mathrm{KB} \cup \mathrm{Assumpties} \vDash \mathrm{OBS}$
  - Probabilistisch redeneren: $P(\mathrm{E} = e \mid \mathrm{OBS})$

Intelligent Systeem als Logische Theorie



Geschiktheid van Logica



Representatie van een multiplier-adder:

$\forall x((\mathsf{MUL}(x) \rightarrow \mathrm{in}_1(x) \times \mathrm{in}_2(x) = \mathrm{out}(x))$
$\forall x((\mathsf{ADD}(x) \rightarrow \mathrm{in}_1(x) + \mathrm{in}_2(x) = \mathrm{out}(x))$

$\mathrm{MUL}(M_1), \mathrm{MUL}(M_2), \mathrm{MUL}(M_3), \mathrm{ADD}(A_1), \mathrm{ADD}(A_2)$
$\mathrm{in}_1(A_1) = \mathrm{out}(M_1), \mathrm{in}_2(A_1) = \mathrm{out}(M_2), \mathrm{in}_1(A_2) = \mathrm{out}(M_2),$
$\mathrm{in}_2(A_2) = \mathrm{out}(M_3), \mathrm{in}_1(M_1) = 3, \mathrm{in}_2(M_1) = 2, \mathrm{in}_1(M_2) = 3,$
$\mathrm{in}_2(M_2) = 2, \mathrm{out}(A_1) = \mathrm{out}(A_2) = 12$

Logisch Redeneren

KB: logische kennisbank en F: feiten

Oplossingsmethoden:

- Deductieve oplossing S: $\mathrm{KB} \cup F \vDash S$, met $\mathrm{KB} \cup F \nvDash \bot$,
- Abductie en inductie:

$$\mathrm{KB} \cup S \cup K \vDash F,$$

  met K contextuele kennis, dan
  - Abductie: KB bevat causale kennis
  - Inductie: KB is achtergrondkennis en S is een inductieve oplossing (geleerde regels)
- Consistentie-gebaseerde oplossing: $\mathrm{KB} \cup S \cup F \nvDash \bot$

Crash Course in Propositielogica

- Welgevormde formules FORM: *atomen* P, *literals* P en $\neg P$, *disjunctie* $(P \vee Q)$, *conjunctie* $(P \wedge Q)$, *implicatie* $(P \rightarrow Q)$
- Waardering (valuatie) of interpretatie w:

$$w : \text{FORM} \rightarrow \{\textit{true}, \textit{false}\}$$

- Model van formule φ: waardering w met $w(\varphi) = \textit{true}$; Notatie: $\boxed{\vDash_w \varphi}$

Crash Course in Propositielogica

- Vervulbaarheid (satisfiability): een formule φ is vervulbaar als er een w bestaat, zodanig dat $\vDash_w \varphi$
- Onvervulbaarheid (unsatisfiability): $\forall w : \nvDash_w \varphi$
- Logisch (semantisch) gevolg (logical consequence): $\forall w$: als $\vDash_w \varphi$ dan $\vDash_w \psi$; Notatie: $\boxed{\varphi \vDash \psi}$
- Lege clause (hokje): als φ onvervulbaar is, dan $\varphi \vDash \Box$ (ook $\varphi \vDash \bot$)

Voorbeeld

"Omdat de VS te veel geld uitgeeft (S) èn banken zuinig zijn (Z), is Nederland in recessie (R)"

Formeel: $((S \wedge Z) \rightarrow R)$

- Met $w(S) = w(Z) = w(R) = \textit{true}$, en dus $w(((S \wedge Z) \rightarrow R)) = \textit{true}$, of $\vDash_w ((S \wedge Z) \rightarrow R)$
- Ook: $((S \wedge Z) \wedge ((S \wedge Z) \rightarrow R)) \vDash R$; andere (verzamelings)notatie: $\{S, Z, (S \wedge Z) \rightarrow R\} \vDash R$
- Met $w'(S) = w'(Z) = \textit{true}$ en $w'(R) = \textit{false}$:

$$\{S, Z, \neg R, (S \wedge Z) \rightarrow R\} \vDash \Box$$

- Korte notatie: we laten haakjes weg ...

Syntactisch: Inferentieregels

- In plaats van met waarderingen (interpretaties) kunnen we ook syntactisch redeneren (let op de vorm ...).

  Voorbeelden:
  - modus ponens regel: $\frac{\varphi, \varphi \to \psi}{\psi}$
  - $\wedge$-introductieregel: $\frac{\varphi, \psi}{\varphi \wedge \psi}$
- Vervang nu $\models$ door syntactische manipulatie (afleidingsrelatie) $\vdash$

  Voorbeeld: Gegeven: P, Q en $(P \wedge Q) \to R$, dan
  - $\{P, Q\} \vdash P \wedge Q$ ($\wedge$-introductie)
  - $\{P, Q, (P \wedge Q) \to R\} \vdash R$ ($\wedge$-introductie en modus ponens)

Soundness & Completeness

- Soundness (gezondheid, correctheid): de inferentieregels respecteren de semantiek, d.w.z. $\boxed{\text{als } \varphi \vdash \psi \text{ dan } \varphi \models \psi}$

  Voorbeeld: $\{P, Q, (P \wedge Q) \to R\} \vdash R$ via $\wedge$-introductie en modus ponens, ook $\{P, Q, (P \wedge Q) \to R\} \models R$
- Completeness (volledigheid): als wat volgt uit een verzameling formules φ, is ook afleidbaar $\boxed{\text{als } \varphi \models \psi \text{ dan } \varphi \vdash \psi}$

  Voorbeeld: $\{P, Q, (P \wedge Q)\ R\} \models R$, en via $\wedge$-introductie en modus ponens: $\{P, Q, (P \wedge Q) \to R\} \vdash R$
- Automatisch gebruik van inferentieregels: er zijn er zoveel!!

Resolutie

- Redeneer alleen met formules in clausevorm:

$$L_1 \vee L_2 \vee \cdots \vee L_n$$

  met L_i een literal, d.w.z. een atoom of negatie van een atoom; als $n = 0$, dan is de clause $\square$ (lege clause)
- Complementaire literals: L en L', zodanig dat $L \equiv \neg L'$
- Resolutie(regel) $\mathcal{R}$ (J.A. Robinson, 1965):

$$\frac{C \vee L, \quad C' \vee \neg L}{D}$$

  met C, C' clauses, en D de (binaire) resolvente gelijk aan $C \vee C'$ met verwijdering van dubbele literals

Voorbeelden Resolutie

- Gegeven $V = \{P \vee Q \vee \neg R, U \vee \neg Q\}$, dan

$$\frac{P \vee Q \vee \neg R, \quad U \vee \neg Q}{P \vee U \vee \neg R}$$

  dus $V \vdash_{\mathcal{R}} P \vee U \vee \neg R$ na eenmalig toepassen van de resolutieregel $\mathcal{R}$
- Als $V \vdash_{\mathcal{R}} \square$ dan heet V inconsistent, en de afleiding heet dan een refutatie; anders, als $V \nvdash_{\mathcal{R}} \square$, dan heet V consistent
- Gegeven $V = \{\neg P \vee Q, \neg Q, P\}$, dan $\frac{\neg P \vee Q, \ \neg Q}{\neg P}$ $\frac{\neg P, \ P}{\square}$

  dus $V \vdash_{\mathcal{R}} \square$, V is inconsistent (als $\vdash_{\mathcal{R}}$ sound is, dan is V ook onvervulbaar)

Soundness & Completeness Resolutie

- Stelling: resolutie is sound (gezond, correct) (dus, $V \vdash_{\mathcal{R}} C \Rightarrow V \models C$) met clause C.

  Bewijs. $C_1 = L \vee C_1'$ en $C_2 = \neg L \vee C_2'$, verder:

  $$\{C_1, C_2\} \vdash_{\mathcal{R}} D$$

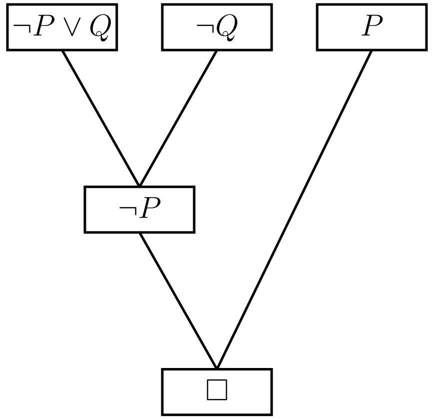
  met D gelijk aan $C_1' \vee C_2'$ na verwijdering van dubbele literals. Te bewijzen:

  $$\models_w (C_1 \wedge C_2) \ \Rightarrow \ \models_w D$$

  voor elke w. Verder: L of $\neg L$ is waar voor w. Stel L is waar, dan moet C_2' waar zijn, dus D is waar Analoog voor $\neg L$ is waar.

- Stelling: resolutie is refutatievolledig (refutation complete) (dus, $V \models \square \Rightarrow V \vdash_{\mathcal{R}} \square$)

Resolutie(refutatie)boom



Gegeven $V = \{\neg P \vee Q, \neg Q, P\}$, dan $V \vdash_{\mathcal{R}} \square$
Opmerking: resolutiebomen zijn *niet* uniek

Resolutie en Predicatenlogica

- Probleem: gegeven

  $S = \{\neg\text{Vader}(\text{alex}, \text{amalia}), \forall x \forall y(\neg\text{Vader}(x, y) \rightarrow \neg\text{Ouder}(x, y))\}$

  We weten $S \models \neg\text{Ouder}(\text{alex}, \text{amalia})$
- Zet om naar clausevorm (resolutie?):

  $S' = \{\neg\text{Vader}(\text{alex}, \text{amalia}), \text{Vader}(x, y) \vee \neg\text{Ouder}(x, y)\}$
- Oplossing: vul voor x in 'alex' en voor y 'amalia'
  $\Rightarrow$ substitutie $\sigma = \{\text{alex}/x, \text{amalia}/y\}$
- Toepassing van resolutie:

  $$\frac{\neg\text{Vader}(\text{alex}, \text{amalia}), \{\text{Vader}(x, y) \vee \neg\text{Ouder}(x, y)\}\sigma}{\neg\text{Ouder}(\text{alex}, \text{amalia})}$$

  dus $S' \vdash_{\mathcal{R}} \neg\text{Ouder}(\text{alex}, \text{amalia})$

Substitutie

- Een substitutie σ is een eindige verzameling van de vorm $\sigma = \{t_1/x_1, \ldots, t_n/x_n\}$, met x_i een variabele en t_i een term; $x_i \neq t_i$ en $x_i \neq x_j$, $i \neq j$
- $E\sigma$ een expressie die uit E verkregen wordt door gelijktijdig alle voorkomens van de variabelen x_i te vervangen door de termen t_i. $E\sigma$ heet een instantiatie
- Als $E\sigma$ geen variabelen bevat, dan wordt $E\sigma$ een basisinstantiatie (ground instance) genoemd

Voorbeelden voor $C = P(x, y) \vee Q(y, z)$:

- $\sigma_1 = \{a/x, b/y\}$, $\sigma_2 = \{y/x, x/y\}$: $C\sigma_1 = P(a, b) \vee Q(b, z)$ en $C\sigma_2 = P(y, x) \vee Q(x, z)$
- $\sigma_3 = \{f(y)/x, g(b)/z\}$: $C\sigma_3 = P(f(y), y) \vee Q(y, g(b))$

Het Gelijkmaken

- Vergelijk ¬Vader(alex, amalia) en Vader(x, y). Wat zijn de verschillen en overeenkomsten?
  - complementaire tekens
  - zelfde predicaatsymbool ('Vader')
  - constante 'alex' versus variabele x en constante 'amalia' versus variabele y

  Gelijkmaken via substitutie

$$\sigma = \{\text{alex}/x, \text{amalia}/y\}$$

- Vergelijk $P(x, f(x))$ en $\neg P(g(a), f(g(a)))$; na weglaten tekens, gelijkmaken via

$$\sigma = \{g(a)/x\}$$

Unificatie

- Zij $\theta = \{t_1/x_1, \ldots, t_m/x_m\}$ en $\sigma = \{s_1/y_1, \ldots, s_n/y_n\}$, dan is de compositie, notatie $\theta \circ \sigma$ of $\theta\sigma$, gedefinieerd als:

$$\{t_1\sigma/x_1, \ldots, t_m\sigma/x_m, s_1/y_1, \ldots, s_n/y_n\}$$

  waarbij elk element $t_i\sigma/x_i$ verwijderd wordt waarvoor $x_i = t_i\sigma$ en tevens elk element s_j/y_j waarvoor $y_j \in \{x_1, \ldots, x_m\}$
- Een substitutie σ heet een unifier van E en E' als $E\sigma = E'\sigma$; E en E' heten dan unificeerbaar
- Een unifier θ van expressies E en E' heet de most general unifier (mgu) dan en slechts dan als voor elke unifier σ van E en E' een substitutie λ bestaat, zodanig dat $\sigma = \theta \circ \lambda \Rightarrow$ maak zo algemeen (met variabelen) mogelijke expressies

Voorbeelden unifiers

Zij gegeven de volgende logische expressies

$$R(x, f(a, g(y)))$$

en

$$R(b, f(z, w))$$

Enkele mogelijke unifiers:

- $\sigma_1 = \{b/x, a/z, g(c)/w, c/y\}$
- $\sigma_2 = \{b/x, a/z, f(a)/y, g(f(a))/w\}$
- $\sigma_3 = \{b/x, a/z, g(y)/w\}$ (mgu)

Merk op:

- $\sigma_1 = \sigma_3 \circ \{c/y\}$
- $\sigma_2 = \sigma_3 \circ \{f(a)/y\}$

Resolutie in Predicatenlogica

- Beschouw: $\{C_1 = P(x) \vee Q(x), C_2 = \neg P(f(y)) \vee R(y)\}$; $P(x)$ en $P(f(y))$ zijn *niet* complementair, wel unificeerbaar, bijvoorbeeld $\sigma = \{f(a)/x, a/y\}$
- Resultaat: $C_1\sigma = P(f(a)) \vee Q(f(a))$ en

$$C_2\sigma = \neg P(f(a)) \vee R(a)$$

  $P(f(a))$ en $\neg P(f(a))$ zijn complementair

$$\{C_1\sigma, C_2\sigma\} \vdash_{\mathcal{R}} Q(f(a)) \vee R(a)$$

- Met de mgu $\theta = \{f(y)/x\}$

$$\{C_1\theta, C_2\theta\} \vdash_{\mathcal{R}} Q(f(y)) \vee R(y)$$

Resolutie in Predicatenlogica

- Notatie: als L een literal is, dan is $[L]$ het atoom
- Zij gegeven de twee clauses $C_1 = C_1' \vee L_1$ en $C_2 = C_2' \vee L_2$, met L_1 positief, en L_2 negatief (negatie)
- Stel $[L_1]\sigma = [L_2]\sigma$, met σ een mgu
- Binaire resolutieregel $\mathcal{B}$ voor predicatenlogica:

$$\frac{(C_1' \vee L_1)\sigma, \ (C_2' \vee L_2)\sigma}{C_1'\sigma \vee C_2'\sigma}$$

$C_1'\sigma \vee C_2'\sigma$ is binaire resolvente, en

$$\{C_1, C_2\} \vdash_{\mathcal{B}} C_1'\sigma \vee C_2'\sigma$$

Factoriseren

- Als de literals $L_1, \ldots, L_k$, $k \geq 2$, in een clause C de most general unifier σ hebben, dan heet de clause $C\sigma$ een factor van C
- Waarvoor nodig? Zij gegeven de clause

$$P(g(x), h(y)) \vee Q(z) \vee P(v, h(a))$$

met

$$\sigma = \{g(x)/v, a/y\}$$

de most general unifier van $P(g(x), h(y))$ en $P(v, h(a))$, dan is $P(g(x), h(a)) \vee Q(z) \vee P(g(x), h(a)) \equiv P(g(x), h(a))$ $\vee Q(z)$ een factor van C

- Nodig voor verwijderen van dubbele literals

Resolutie in Predicatenlogica

Een resolvente van de (algemene) resolutieregel $\mathcal{R}$ in de predikatenlogica is:

- een binaire resolvente van C_1 en C_2, òf
- een binaire resolvente van C_1 en een factor van C_2, òf
- een binaire resolvente van een factor van C_1 en C_2, òf
- een binaire resolvente van een factor van C_1 en een factor van C_2

Notatie:

$$S \vdash_{\mathcal{R}} C$$

afleiding (eventueel refutatie) van C uit de verzameling clauses S in nul of meer stappen

Algemene Syntax naar Clausevorm

Resolutie snapt alleen clausevorm $\Rightarrow$ formules in predicatenlogica vertalen naar clausevorm, via:

- Schuif quantoren naar buiten (prenex-normaalvorm):

$$Q_1 x_1 \cdots Q_n x_n M$$

  met elke $Q_i \in \{\forall, \exists\}$, en M is een quantorvrije formule $Q_1 x_1 \ldots Q_n x_n$ is de prefix en M is de matrix
- Skolem-normaalvorm: vervang $\exists x$ door termen met functiesymbolen van voorafgaande universeel gekwantificeerde variabelen (skolemiseren) $\forall x \exists y P(x, y)$ wordt $\forall x P(x, f(x))$
- Een formule φ in skolem prenex-normaalvorm is in conjunctieve normaalvorm (clausevorm) als de matrix van de vorm $C_1 \wedge \cdots \wedge C_n$ met elke C_i een clause

Missionarissen en Kannibalen

Drie missionarissen en drie kannibalen staan aan de westerlijke oever van een rivier. Er ligt een boot op de westerlijke oever waarmee men de rivier kan overvaren; de boot kan niet meer dan twee personen bevatten. De drie missionarissen willen de rivier oversteken naar de oosterlijke oever. Er is echter een probleem: als op één van de oevers meer kannibalen dan missionarissen zijn, zullen de kannibalen de missionarissen opeten.

Bestaat er een manier voor de missionarissen om, zonder verlies van levens, op de oosterlijke oever te komen?

Kennisrepresentatie

- Een kannibaal vaart met de boot van de westerlijke oever naar de oosterlijke oever, of omgekeerd:

$$\forall x \forall y \forall z \forall w$$
$$(\mathit{Toestand}(\mathit{west}(m(x), c(s(y))), \mathit{boot\_west}, \mathit{oost}(m(z), c(w)))$$
$$\leftrightarrow$$
$$\mathit{Toestand}(\mathit{west}(m(x), c(y)), \mathit{boot\_oost}, \mathit{oost}(m(z), c(s(w)))))$$

- Twee missionarissen varen met de boot van de westerlijke oever naar de oosterlijke oever, of omgekeerd:

$$\forall x \forall y \forall z \forall w$$
$$(\mathit{Toestand}(\mathit{west}(m(s(s(x))), c(y)), \mathit{boot\_west}, \mathit{oost}(m(z), c(w)))$$
$$\leftrightarrow$$
$$\mathit{Toestand}(\mathit{west}(m(x), c(y)), \mathit{boot\_oost}, \mathit{oost}(m(s(s(z))), c(w))))$$

- etcetera

Toepassing van ARP (Otter)

```
---------------- PROOF ----------------
2 [] Toestand(west(m(x),c(s(y))),boot_west,
         oost(m(z),c(w)))|
     -Toestand(west(m(x),c(y)),boot_oost,
          oost(m(z),c(s(w)))).
3 [] -Toestand(west(m(s(s(x))),c(y)),boot_west,
          oost(m(z),c(w)))|
     Toestand(west(m(x),c(y)),boot_oost,
         oost(m(s(s(z))),c(w))).
6 [] Toestand(west(m(s(x)),c(s(y))),boot_west,oost(m(z),c(w)))|
     -Toestand(west(m(x),c(y)),boot_oost,oost(m(s(z)),c(s(w)))).
9 [] -Toestand(west(m(x),c(s(s(y)))),boot_west,
          oost(m(z),c(w)))|
     Toestand(west(m(x),c(y)),boot_oost,
         oost(m(z),c(s(s(w))))).
17 [] Toestand(west(m(s(s(s(0)))),c(s(s(s(0))))),
          boot_west,oost(m(0),c(0))).
18 [] -Toestand(west(m(0),c(x)),y,oost(m(s(s(s(0)))),c(z))).
```


Vervolg ...

Stap 1: 3 missionarissen en 1 kannibaal west

```
19 [17,9] Toestand(west(m(s(s(s(0)))),c(s(0))),
                   boot_oost,oost(m(0),c(s(s(0))))).
22 [19,2] Toestand(west(m(s(s(s(0)))),c(s(s(0)))),
             boot_west,oost(m(0),c(s(0)))).
23 [22,9] Toestand(west(m(s(s(s(0)))),c(0)),
             boot_oost,oost(m(0),c(s(s(s(0)))))).
24 [23,2] Toestand(west(m(s(s(s(0)))),c(s(0))),
             boot_west,oost(m(0),c(s(s(0))))).
25 [24,3] Toestand(west(m(s(0)),c(s(0))),
             boot_oost,oost(m(s(s(0))),c(s(s(0))))).
26 [25,6] Toestand(west(m(s(s(0))),c(s(s(0)))),
             boot_west,oost(m(s(0)),c(s(0)))).
27 [26,3] Toestand(west(m(0),c(s(s(0)))),
             boot_oost,oost(m(s(s(s(0)))),c(s(0)))).
28 [27.1,18.1] $F.
------------ end of proof -------------
```

Eind: 3 missionarissen en 1 kannibaal oost