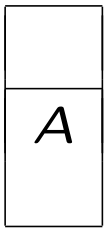
Discourse
Slides 20-2-2009
Henk Zeevat

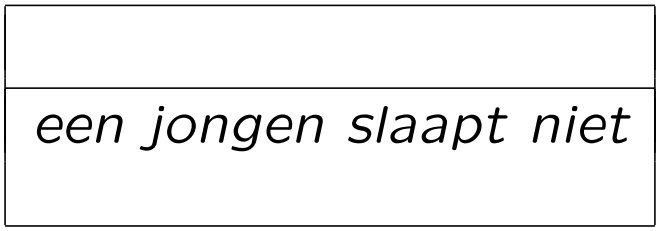
Een jongen slaapt niet.

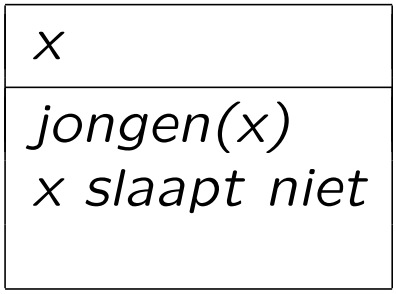
Nieuwe regel

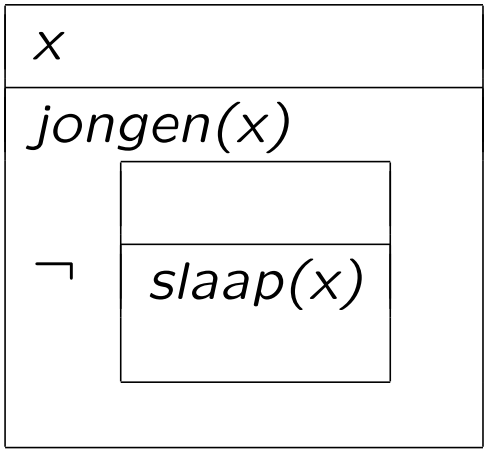
niet A



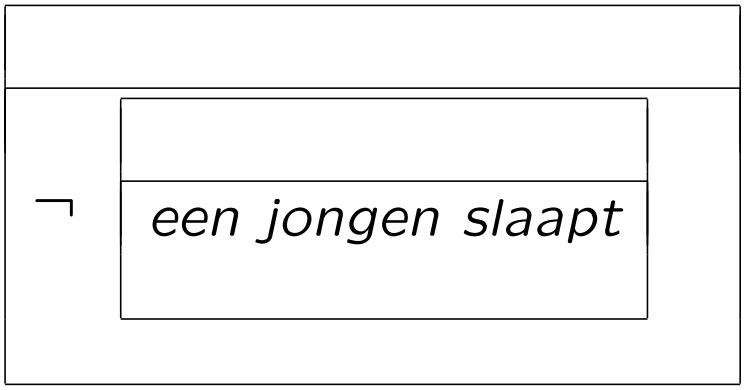
$M \models \neg K[f]$ desda er is geen g $f \subseteq g$ en $dom(g) = dom(f) \cup K_0$ and $\forall \varphi \in K_1$ $M \models \varphi[g]$

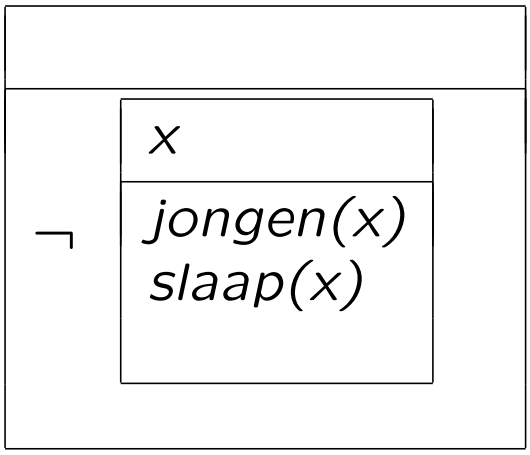

een jongen slaapt niet


x
jongen(x)
x slaapt niet


x
jongen(x)
¬
slaap(x)

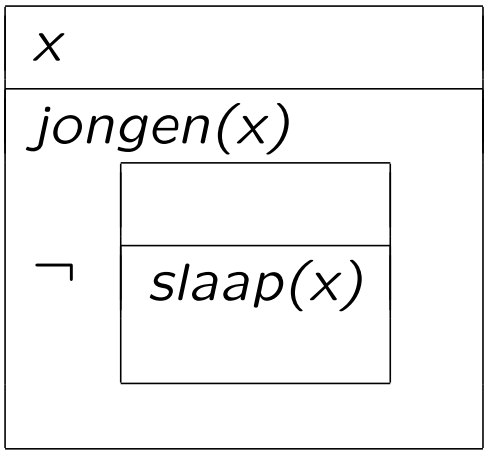
Of anders



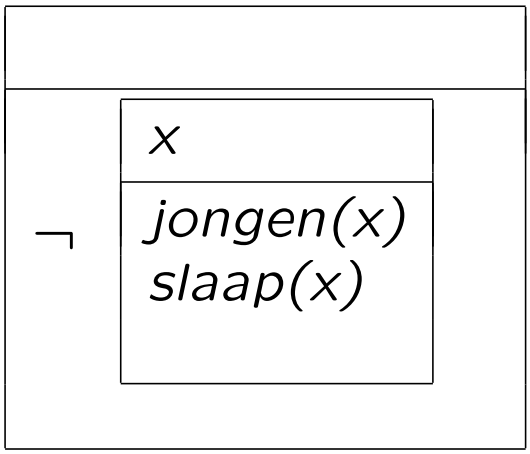


Model M

$A = \{a, b\}$
$F(jongen) = \{a, b\}$
$F(slaap) = \{a\}$



waar: kies $f(x) = b$



niet waar

f is de lege functie

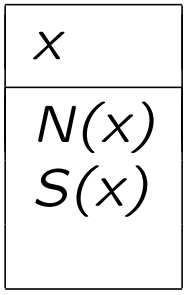
Kies $g(x) = a$

Jan heeft geen auto.

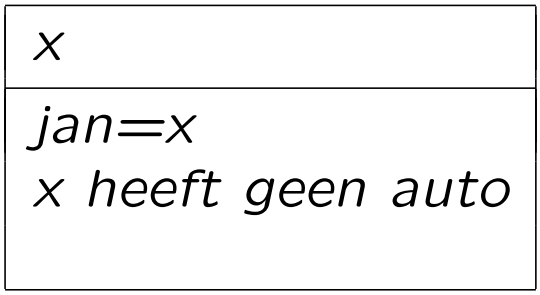
nieuwe regel:

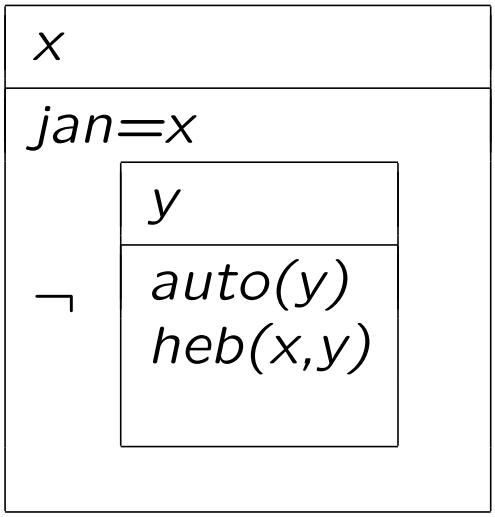
S geen N

nieuwe discourse referent x

$\neg$ 


Jan heeft geen auto


x
jan=x
x heeft geen auto


x
jan=x
¬
y
auto(y)
heb(x,y)

Opmerking:

$K \Rightarrow K' =_{def} \neg(K \cup \{\neg K'\})$

$(K \cup K' =< K_0 \cup K'_0, K_1 \cup K'_1 >)$

Geeft de goede semantiek (opgave)

Maakt de toegankelijkheidsrelatie duidelijker.

Toegankelijkheid

Pronomina vinden een oude discourse referent die *toegankelijk* is

Jan heeft geen auto. Hij staat in de garage.

Iedere jongen slaapt. Hij snurkt.

Als een boer een ezel heeft slaat hij hem. Hij is erg ongelukkig.

Definitie

1. K heeft toegang to K
2. K'' in $K' \Rightarrow K''$ heeft toegang tot K'
3. Als $K' \Rightarrow K''$ een conditie van K is heeft K' toegang tot K
4. Als $\neg K'$ een conditie is van K heeft K' toegang tot K
5. Als K'' toegang heeft tot K' en K' tot K dan heeft K'' toegang tot K

Correspondeert met de domeinen van de functies

x is toegankelijk als f(x) gedefinieerd is voor een functie die de locale DRS kan vervullen als onderdeel van de vervulling van de hele DRS

Relatie met predikaatlogica

$trans(K)$

$trans(K) = \exists K_0 \bigwedge trans1(K_1)$

$trans1(K_1) : \{trans2(\varphi) : \varphi \in K_0\}$

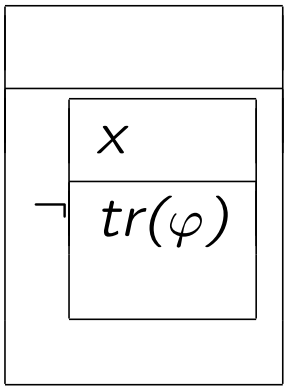
$trans2(\varphi) = \varphi$ als φ atomair is

$trans2(\neg K) : \forall K_0 \neg \bigwedge trans1(K_1)$

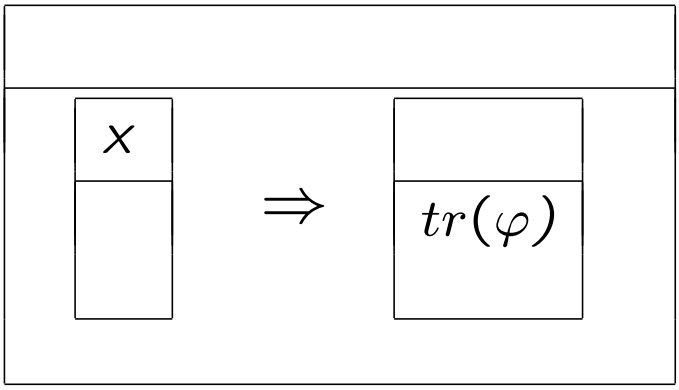
$K \Rightarrow K' : \forall K_0 (\bigwedge trans1(K_1) \rightarrow \exists K'_0 \bigwedge trans1(K'_1))$

Omgekeerd:

$tr(\exists x\varphi) = \neg$

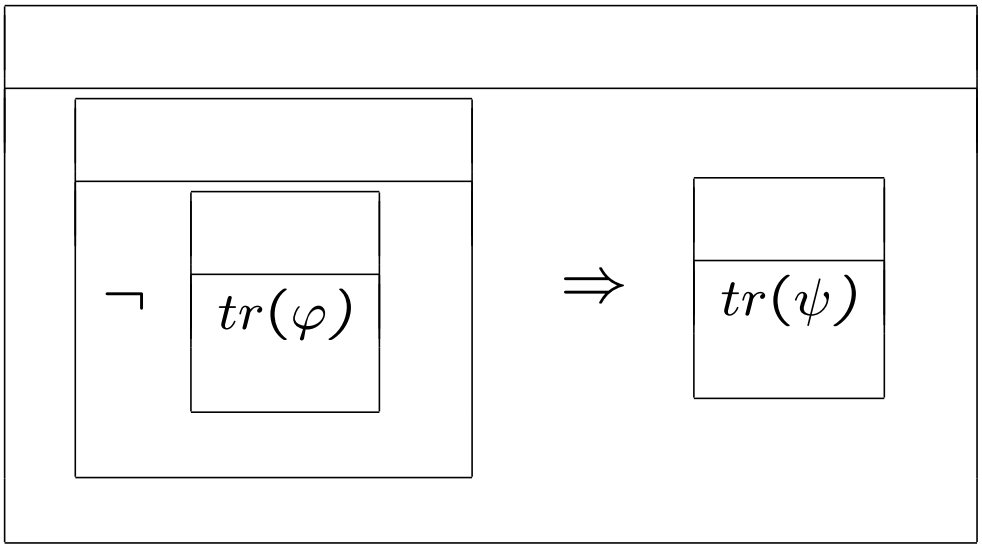


$tr(\forall x\varphi) =$

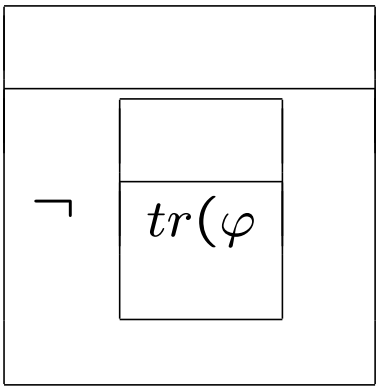


$tr(\varphi \wedge \psi) =< tr(\varphi)_0 \cup tr(\psi)_0, \varphi)_1 \cup tr(\psi)\_1 >$

$tr(\varphi \vee \psi) =$



$tr(\neg\varphi) =$



Merk op: wel equivalent (zelfde abstracte formules), maar zin voor zin vertaling werkt niet.

Eigennamen en presuppositie

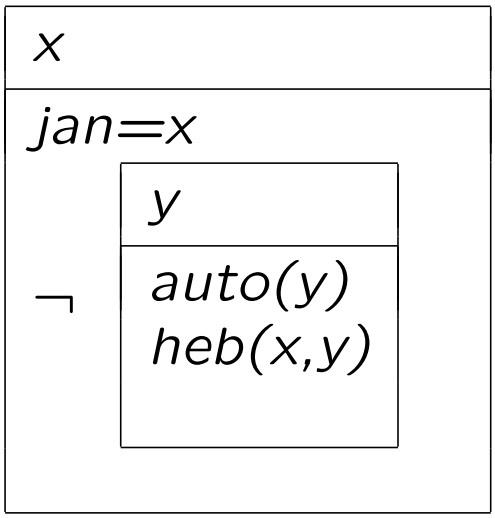
We hadden de regel:

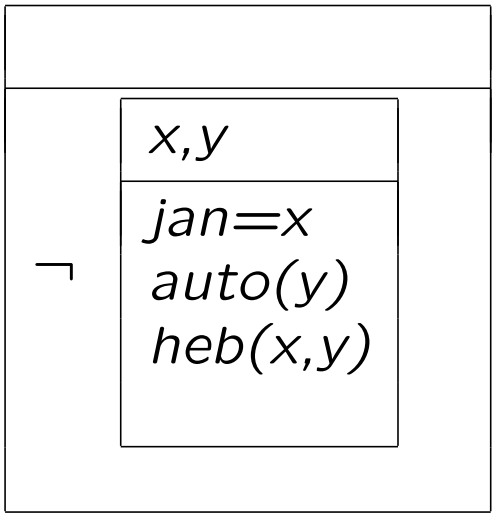
namen: Jan, c

nieuwe discourse referent: x

nieuwe conditie: $c = x$

vervang naam door x


x
jan=x
y
¬
auto(y)
heb(x,y)


x,y
¬
jan=x
auto(y)
heb(x,y)

Problemen:

a. Niemand heet Jan

b. Twee mensen die Jan heten.

c. “Jan” moet opgelost worden

Andere ontwikkelingsregel: maak een subdrs:

x
name(x, "jan")

op de plaats van de naam.

probeer een stuk van de toegankelijke DRS te vinden met een discourse referent z zodat $name(z,''jan'')$ ("jan" is een naam van x) het geval is en gebruik dan z voor de substitutie

als dat niet lukt:

voeg x en $name(x,''jan'')$ op het hoogste niveau van de DRS

Dit is een byzonder geval van het algemene mechanisme voor presuppositieoplossing.

Ander geval:

de rode bal

maak een subdrs:

x
rood(x), bal(x)

op de plaats van de naam.

probeer een stuk van de toegankelijke DRS te vinden met een discourse referent z zodat $rood(x)$ en $bal(x)$ het geval is en gebruik dan z voor de substitutie

als dat niet lukt:

voeg x en $rood(x)$ en $bal(x)$ toe op het hoogste niveau van de DRS

Opgave: vind een gebruikswijze van "de N" waar dit niet voor werkt.