What syntax doesn’t feed semantics
Fake indexicals as indexicals

Emar Maier

ILLC/University of Amsterdam

www.ru.nl/ncs/~emar

ESSLLI Workshop ‘What syntax feeds semantics?’ , Hamburg
August 11, 2008
Introduction: /

- Kaplan (1977, 1989):
  - / is indexical, like *today*
    1. context-dependent
    2. directly referential
  - 2D semantics

I am speaking $\not\equiv$ the speaker is speaking

  - / is a pronoun, like *he*
  - pronouns have bound and referential readings

Only I did my homework

sloppy others didn’t do theirs: $\forall x[ x \neq i \rightarrow \neg \text{do hw}(x, x)]$

- today: defend Kaplan
Introduction: I

- Kaplan (1977, 1989):
  - I is indexical, like today
    1. context-dependent
    2. directly referential
  - 2D semantics

\[ \text{I am speaking} \neq \text{the speaker is speaking} \]

  - I is a pronoun, like he
  - pronouns have bound and referential readings

Only I did my homework

\[ \text{sloppy others didn’t do theirs: } \forall x [x \neq i \rightarrow \neg \text{do.hw}(x, x)] \]

- today: defend Kaplan
Introduction: /

- Kaplan (1977, 1989):
  - I is indexical, like today
    1. context-dependent
    2. directly referential
  - 2D semantics

  - I is a pronoun, like he
  - pronouns have bound and referential readings

I am speaking \( \not\equiv \) the speaker is speaking

Only I did my homework

sloppy others didn’t do theirs: \( \forall x[ x \neq i \rightarrow \neg do\cdot hw(x, x)] \)

- today: defend Kaplan
Introduction: I

- Kaplan (1977, 1989):
  - I is indexical, like *today*
    1. context-dependent
    2. directly referential
  - 2D semantics

  I am speaking \(\neq\) the speaker is speaking

  - I is a pronoun, like *he*
  - pronouns have bound and referential readings

Only I did *my* homework

sloppy others didn’t do theirs: \(\forall x [x \neq i \rightarrow \neg do.hw(x, x)]\)

- today: defend Kaplan
Introduction

Fake indexicals

Pronouns in Generative Linguistics
VP ellipsis
Only

Ellipsis resolution by unification
Only by unification
Avoiding sloppy names

De se binding and de re acquaintance
Avoiding de se names

Outline

1 Introduction

2 Fake indexicals
   Pronouns in Generative Linguistics
   VP ellipsis
   Only

3 ... as indexicals
   Ellipsis resolution by unification
   Only by unification
   Avoiding sloppy names

4 De se binding and de re acquaintance
   Avoiding de se names

Emar Maier
What syntax doesn’t feed semantics Fake indexicals as indexicals
Outline

1 Introduction

2 Fake indexicals
   Pronouns in Generative Linguistics
   VP ellipsis
   Only

3 ... as indexicals
   Ellipsis resolution by unification
   Only by unification
   Avoiding sloppy names

4 De se binding and de re acquaintance
   Avoiding de se names
The syntax-semantics interface
The syntax-semantics interface
The syntax-semantics interface
Binding and coreference

SS: John did his homework
Binding and coreference

SS: John did his homework

LF: [John did his homework]
Binding and coreference

SS: John did his homework

LF: [John did his homework]

$L$: do.homework.of$(j, x)$
**Binding and coreference**

**SS:** John did his homework

**LF:** [John did his homework]

\[ \downarrow \]
\[ \mathcal{L}: \text{do.homework.of}(j, x) \]
\[ \downarrow \]
\[ m: \left[ \text{do.homework.of}(j, x) \right]^f_w = 1 \]
iff John did homework of \( f(x) \)
Binding and coreference

SS: John did his homework

LF: [John did his homework]

\[ \text{L: do.homework.of}(j, x) \]
\[ \text{m: } [\text{do.homework.of}(j, x)]^f_w = 1 \]
iff John did homework of \( f(x) \)

context: his = \( f(x) = \text{John} \)
Binding and coreference

**SS**: John did his homework

**LF**: [John did his homework]

$$\downarrow$$

$$\mathcal{L}$$: do.homework.of($j, x$)

$$\downarrow$$

$$m: [do.homework.of(j, x)]_w^f = 1$$

iff John did homework of John

context: his = $f(x) = John$
Binding and coreference

SS: John did his homework

LF: [John did his homework]

\[ \lambda x[\text{do.homework.of}(x,x)](j) \]

\[ \text{context: his} = f(x) = \text{John} \]
Binding and coreference

**SS:** John did his homework

**LF:** [John did his homework]  
\[ \downarrow \]  
\[ \mathcal{L}: \text{do.homework.of}(j, x) \]  
\[ \downarrow \]  
\[ m: \left[ \text{do.homework.of}(j, x) \right]_w^f = 1 \]  
iff John did homework of John

context: his = f(x) = John
Binding and coreference

**SS:** John did his homework

**LF:** [John did his homework]

\[ \mathcal{L}: \text{do.homework.of}(j, x) \]

\[ m: \left[ \text{do.homework.of}(j, x) \right]^f_w = 1 \]

iff John did homework of John

context: his = f(x) = John

**John^1 [ t_1 did his_1 homework]**

\[ \downarrow \]

\[ \text{do.homework.of}(j, j) \]

\[ \left[ \text{do.homework.of}(j, j) \right]^f_w = 1 \text{ iff } \]

John did homework of John
VP ellipsis

PF: John likes his dad but Peter doesn’t

- ambiguous:
  - strict: Peter doesn’t like John’s
dad
  - sloppy: Peter doesn’t like his own
VP ellipsis

**PF:** John likes his dad but Peter doesn’t

- ambiguous:
  - strict: Peter doesn’t like John’s
dad
  - sloppy: Peter doesn’t like his own
dad
- Sag/Williams: reduce to referential–bound ambiguity
  - delete an LF constituent at PF if it’s *semantically equivalent* to an earlier constituent at LF
VP ellipsis

**PF:** John likes his dad but Peter doesn’t

- ambiguous:
  - **strict:** Peter doesn’t like John’s
d  
  - **sloppy:** Peter doesn’t like his own

- Sag/Williams: reduce to referential–bound ambiguity
  - delete an LF constituent at PF if it’s *semantically equivalent* to an earlier constituent at LF

**LF:** John likes his dad but
Peter doesn’t like his dad

(context: his = his = John)
VP ellipsis

PF: John likes his dad but Peter doesn’t

- ambiguous:
  - strict: Peter doesn’t like John’s
dirty: Peter doesn’t like his own

- Sag/Williams: reduce to referential-bound ambiguity
  - delete an LF constituent at PF if it’s semantically equivalent to
    an earlier constituent at LF

LF: John likes his dad but
    Peter doesn’t like his dad

(context: his = his = John)
**VP ellipsis**

**PF:** John likes his dad but Peter doesn’t

- ambiguous:
  - **strict:** Peter doesn’t like John’s
  - **sloppy:** Peter doesn’t like his own
- Sag/Williams: reduce to referential–bound ambiguity
  - delete an LF constituent at PF if it’s *semantically equivalent* to an earlier constituent at LF

**LF:** John likes his dad but

Peter doesn’t like his dad

(context: his = his = John)

\[\text{like.dad}(j, x) \land \text{like.dad}(p, y)\]
VP ellipsis

PF: John likes his dad but Peter doesn’t

- ambiguous:
  - strict: Peter doesn’t like John’s
d  sloppy: Peter doesn’t like his own

- Sag/Williams: reduce to referential-bound ambiguity
  - delete an LF constituent at PF if it’s semantically equivalent to an earlier constituent at LF

LF: John likes his dad but
Peter doesn’t like his dad

\(\text{context: } f(x) = f(y) = [j] \)
\(\text{like.dad}(j, x) \land \text{like.dad}(p, y)\)
VP ellipsis

**PF:** John likes his dad but Peter doesn’t

- ambiguous:
  - strict: Peter doesn’t like John’s
d  - sloppy: Peter doesn’t like his own

- Sag/Williams: reduce to referential–bound ambiguity
  - delete an LF constituent at PF if it’s *semantically equivalent* to an earlier constituent at LF

**LF:** John likes his dad but Peter doesn’t like his dad

(/context: \( f(x) = f(y) = [j] \)) \( \lambda x [\text{like.dad}(j, x) \land \text{like.dad}(p, y)] \)
VP ellipsis

PF: John likes his dad but Peter doesn’t

- ambiguous:
  - strict: Peter doesn’t like John’s dad
  - sloppy: Peter doesn’t like his own dad

- Sag/Williams: reduce to referential–bound ambiguity
  - delete an LF constituent at PF if it’s semantically equivalent to an earlier constituent at LF

LF: John likes his dad but Peter doesn’t like his dad

(context: \( f(x) = f(y) = [j] \))
like.dad(j, x) \land like.dad(p, y)

John\(^1\) [\( t_1 \) likes his\(_1\) dad] but Peter\(^1\) [\( t_1 \) doesn’t like his\(_1\) dad]
VP ellipsis

**PF:** John likes his dad but Peter doesn’t

- ambiguous:
  - strict: Peter doesn’t like John’s
  - sloppy: Peter doesn’t like his own

- Sag/Williams: reduce to referential–bound ambiguity
  - delete an LF constituent at PF if it’s *semantically equivalent* to an earlier constituent at LF

**LF:** John likes his dad but Peter doesn’t like his dad

\[(\text{context: } f(x) = f(y) = [j]) \]
\[\text{like.dad}(j, x) \land \text{like.dad}(p, y)\]
VP ellipsis

**PF:** John likes his dad but Peter doesn’t

- ambiguous:
  - strict: Peter doesn’t like John’s
  - sloppy: Peter doesn’t like his own

- Sag/Williams: reduce to referential–bound ambiguity
  - delete an LF constituent at PF if it’s *semantically equivalent* to an earlier constituent at LF

**LF:** John likes his dad but Peter doesn’t like his dad

(context: f(x)=f(y)=\lbrack j \rbrack)

like.dad(j, x) \land like.dad(p, y)

John^1 [t_1 \text{ likes } \text{ his}_1 \text{ dad}] \text{ but } Peter^1 [t_1 \text{ doesn’t like } \text{ his}_1 \text{ dad}]

\text{like.dad}(j, j) \land \neg \text{like.dad}(p, p)
Binding an indexical?

**PF:** I like my job, but Sue doesn’t
Binding an indexical?

**PF:** I like my job, but Sue doesn’t

**LF:** I like my job but Sue doesn’t like my job
Binding an indexical?

**PF:** I like my job, but Sue doesn’t

**LF:** I like my job but
Sue doesn’t like my job

like.job(i, i) \land \neg\ like.job(s, i)
**PF:** I like my job, but Sue doesn’t

**LF:** I like my job but
Sue doesn’t like my job

\[
\text{like.job}(i, i) \land \neg \text{like.job}(s, i)
\]

**I^1 [t_1 like my_1 job] but**
**Sue^1 [t_1 doesn’t like my_1 job]**

**Introduction**

Fake indexicals... as indexicals

*De se binding and de re acquaintance*
Binding an indexical?

**PF:** I like my job, but Sue doesn’t

**LF:** I like my job but
Sue doesn’t like my job

\[ \text{like.job}(i, i) \land \neg \text{like.job}(s, i) \]

\[ [t_1 \text{ like my}_1 \text{ job}] \text{ but } [t_1 \text{ doesn’t like my}_1 \text{ job}] \]

\[ \lambda x[\text{like.job}(x, x)](i) \land
\neg \lambda x[\text{like.job}(x, x)](s) \]
 Binding an indexical?

**PF:** I like my job, but Sue doesn’t

**LF:** I like my job but
Sue doesn’t like my job

\[ \text{like.job}(i, i) \land \neg \text{like.job}(s, i) \]

\[ \text{like}.\text{job}(i, i) \land
\neg \text{like}.\text{job}(s, s) \]
PF: Only I did my homework
只有我做了作业

\[ \text{只有我}^{1} \ [t_{1} \text{ 做了作业}] \]
Only

**PF:** Only I did my homework

\[
\text{only}(i)(\lambda x[\text{do.hw}(x, i)])
\]

\[
[\text{Only I}]^1 [t_1 \text{ did my homework}]
\]
Only

**PF**: Only I did my homework

\[\text{only}(i)(\lambda x[\text{do.hw}(x, i)])\]

\[\text{only}(x)(P) \equiv \forall y [y \neq x \rightarrow \neg P(y)]\]
**PF:** Only I did my homework

\[
\text{only}(i)(\lambda x[\text{do.hw}(x, i)])
\]

\[
\text{only}(x)(P) \equiv \forall y[y \neq x \rightarrow \neg P(y)]
\]
PF: Only I did my homework

\[ [\text{Only I}]^1 [t_1 \text{ did my homework}] \]
\[ \text{only}(i)(\lambda x[\text{do.hw}(x, i)]) \]

\[ [\text{Only I}]^1 [t_1 \text{ did my}_1 \text{ homework}] \]
\[ \text{only}(i)(\lambda x[\text{do.hw}(x, x)]) \]
\[ \text{only}(x)(P) \equiv \forall y[y \neq x \rightarrow \neg P(y)] \]
Outline

1. Introduction
2. Fake indexicals
   - Pronouns in Generative Linguistics
   - VP ellipsis
   - Only
3. ... as indexicals
   - Ellipsis resolution by unification
   - Only by unification
   - Avoiding sloppy names
4. De se binding and de re acquaintance
   - Avoiding de se names

Emar Maier

What syntax doesn’t feed semantics Fake indexicals as indexicals
Ellipsis by unification

SS: I like my job, but Sue doesn’t
Ellipsis by unification

SS: I like my job, but Sue doesn’t

\[ \text{like}.\text{job}(i, i) \land \neg P(s) \]
Ellipsis by unification

SS: I like my job, but Sue doesn’t

like.job(i, i) \land \neg P(s)
SS: I like my job, but Sue doesn’t

\[
\text{like.job}(i, i) \land \neg P(s) \\
P(i) \equiv \text{like.job}(i, i)
\]
Ellipsis by unification

SS: I like my job, but Sue doesn’t

\[
\text{like.job}(i, i) \land \neg \text{P}(s) \\
\text{P}(i) \models \text{like.job}(i, i)
\]

\[
P \mapsto \lambda x[\text{like.job}(x, i)] \\
P \mapsto \lambda x[\text{like.job}(x, x)]
\]
Ellipsis by unification

SS: I like my job, but Sue doesn’t

\[
\text{like.\(job(i, i) \land \neg P(s)\)}
\]

\[
P(i) \equiv \text{like.\(job(i, i)\)}
\]

\[
P \mapsto \lambda x[\text{like.\(job(x, i)\)}]
\]

\[
\text{like.\(job(i, i) \land \neg \text{like.\(job(s, i)\)}\)}
\]

\[
P \mapsto \lambda x[\text{like.\(job(x, x)\)}]
\]
**Ellipsis by unification**

SS: I like my job, but Sue doesn’t

\[
\begin{align*}
\text{like.job}(i, i) & \land \neg \text{P}(s) \\
P(i) & \equiv \text{like.job}(i, i) \\
\text{P} & \mapsto \lambda x[\text{like.job}(x, i)] \quad \text{P} & \mapsto \lambda x[\text{like.job}(x, x)] \\
\text{like.job}(i, i) & \land \neg \text{like.job}(s, i) \quad \text{like.job}(i, i) & \land \neg \text{like.job}(s, s)
\end{align*}
\]

cf. Dalrymple et al. 1991
Only by unification

SS: Only $[I]_F$ did my homework
Only by unification

SS: Only $[I]_F$ did my homework

$$\forall x [x \neq i \rightarrow \neg B(x)]$$
Only by unification

SS: Only $[I]_F$ did my homework

$$\forall x[x \neq i \rightarrow \neg B(x)]$$

$$B(i) \equiv \text{do.hw}(i, i)$$
Only by unification

SS: Only $[I]_F$ did my homework

\[ \forall x [x \neq i \rightarrow \neg B(x)] \]
\[ B(i) \models \text{do.hw}(i, i) \]

\[ B \mapsto \lambda x [\text{do.hw}(x, i)] \]
\[ B \mapsto \lambda x [\text{do.hw}(x, x)] \]
Only by unification

**SS:** Only $[I]_F$ did my homework

\[
\forall x [x \neq i \rightarrow \neg B(x)]
\]

\[
B(i) \models \text{do.hw}(i, i)
\]

\[
B \mapsto \lambda x [\text{do.hw}(x, i)]
\]

\[
\forall x [x \neq i \rightarrow \neg \text{do.hw}(x, i)]
\]

cf. Pulman (1997)
Conclusions

- semantic/pragmatic alternative:
  - minimized syntactic levels
  - \( I \) is true indexical, interpreted in situ
  - derive strict/sloppy by HOU
Conclusions

- semantic/pragmatic alternative:
  - minimized syntactic levels
  - I is true indexical, interpreted in situ
  - derive strict/sloppy by HOU
- Kaplan saved?
No sloppy names

John likes John’s job but Sue doesn’t

- strict: Sue doesn’t like John’s job

Only Mary likes Mary’s job

- strict: \( \sim \) others don’t like Mary’s
Predictions

- generative:
  - names $\neq$ pronouns
  - Principle C prohibits bound names
Predictions

- generative:
  - names ≠ pronouns
  - Principle C prohibits bound names
  - prediction: only reference, only strict
Predictions

- generative:
  - names ≠ pronouns
  - Principle C prohibits bound names
  - prediction: only reference, only strict

- pragmatic:
  - names ≈ indexicals: directly referential
Predictions

- generative:
  - names \( \neq \) pronouns
  - Principle C prohibits bound names
  - prediction: only reference, only strict

- pragmatic:
  - names \( \approx \) indexicals: directly referential
  - prediction: strict + sloppy (by HOU)
Pragmatic blocking

- competing alternatives:

  (1) Only Mary likes Mary’s job
  (2) Only Mary likes her job

• (1) violates Principle C
• (1) more marked by referential hierarchy: definite descriptions > names > pronouns
• ulterior pragmatic motive for using (1)?
  • topicalizes/presupposes/makes salient Mary’s job
  • prioritizes background containing Mary’s job

B ⇝ { \( \lambda x . \text{like} . \text{job}(x,m) \), \( \lambda x . \text{like} . \text{job}(x,x) \) }
Pragmatic blocking

- competing alternatives:

(1) Only Mary likes Mary’s job
(2) Only Mary likes her job

- (1) violates Principle C
- (1) more marked by referential hierarchy:
  - definite descriptions > names > pronouns
Pragmatic blocking

- competing alternatives:
  
  (1) Only Mary likes Mary’s job
  
  (2) Only Mary likes her job

- (1) violates Principle C
- (1) more marked by referential hierarchy:
  - definite descriptions $>$ names $>$ pronouns
- ulterior pragmatic motive for using (1)?
Pragmatic blocking

• competing alternatives:

(1) Only Mary likes Mary’s job
(2) Only Mary likes her job

• (1) violates Principle C
• (1) more marked by referential hierarchy:
  • definite descriptions > names > pronouns
• ulterior pragmatic motive for using (1)?
  • topicalizes/presupposes/makes salient Mary’s job
    • prioritize background containing Mary’s job
Pragmatic blocking

- competing alternatives:
  
(1) Only Mary likes Mary’s job

(2) Only Mary likes her job

- (1) violates Principle C
- (1) more marked by referential hierarchy:
  - definite descriptions > names > pronouns
- ulterior pragmatic motive for using (1)?
  - topicalizes/presupposes/makes salient Mary’s job
    - prioritize background containing Mary’s job
    - \[ B \rightarrow \{ \lambda x [ \text{like.job}(x, m)], \lambda x [ \text{like.job}(x, x)] \} \]
Conclusions

- generative
  - syntax/semantics: PF, LF, SS, $\mathcal{L}$, m
  - pronouns vs names
    - binding/reference ambiguity: he, she, they, I, you, ...
    - reference: John, Sue, ...
  - in ellipsis, focus, only:
    - reference $\rightarrow$ strict
    - binding $\rightarrow$ sloppy
Conclusions

• generative
  • syntax/semantics: PF, LF, SS, $\mathcal{L}$, m
  • pronouns vs names
    • binding/reference ambiguity: he, she, they, I, you,…
    • reference: John, Sue,…
  • in ellipsis, focus, only:
    • reference $\rightarrow$ strict
    • binding $\rightarrow$ sloppy

• pragmatic
  • semantics/pragmatics: SS, $\mathcal{L}$, m
  • anaphoric vs directly referential
    • anaphoric: he, she, they,…
    • referential: I, you, John, today,…
  • HOU pragmatically derives strict/sloppy
  • sloppy names pragmatically blocked by anaphoric alternative
Outline

1. Introduction

2. Fake indexicals
   - Pronouns in Generative Linguistics
   - VP ellipsis
   - Only

3. ... as indexicals
   - Ellipsis resolution by unification
   - Only by unification
   - Avoiding sloppy names

4. *De se* binding and *de re* acquaintance
   - Avoiding *de se* names
Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV.

I thought I was at a safe distance from the fire.
Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV.

I thought I was at a safe distance from the fire.

“I’m at safe distance from fire.”
De se and de re

Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV

I thought I was at a safe distance from the fire

“I’m at safe distance from fire”

$\text{BEL}_i[\text{safe}(i)]$
Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV.

I thought I was at a safe distance from the fire.

“I’m at safe distance from fire”

\( \text{BEL}_i[\text{safe}(i)] \)

?I thought that I was remarkably calm.

“that guy is remarkably calm”
**De se and de re**

Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV.

I thought I was at a safe distance from the fire

“I’m at safe distance from fire”

\(\text{BEL}_i[\text{safe}(i)]\)

?I thought that I was remarkably calm

“that guy is remarkably calm”

\(\text{BEL}_i[\text{remarkably.calm}(i)]\)
Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV.

I thought I was at a safe distance from the fire

“I’m at safe distance from fire”

\( \text{BEL}_i^* \lambda x [\text{safe}(x)] \)

?I thought that I was remarkably calm

“that guy is remarkably calm”

\( \text{BEL}_i [\text{remarkably.calm}(i)] \)
Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV

I thought I was at a safe distance from the fire

“I’m at safe distance from fire”

\[ \text{BEL}_i^* \lambda x [\text{safe}(x)] \]

?I thought that I was remarkably calm

“that guy is remarkably calm”

\[ \text{BEL}_i^{e \times e t} \langle i, \lambda x [\text{r.calm}(x)] \rangle \]
Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV.

I thought I was at a safe distance from the fire

“I’m at safe distance from fire”

\[
\text{BEL}_i^* \lambda x [\text{safe}(x)]
\]

I thought that I was remarkably calm

“that guy is remarkably calm”

\[
R(i, i) \land \text{BEL}_i^* \lambda x [r.\text{calm}(\gamma y[R(x, y)]]
\]
**De se and de re**

Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV.

I thought I was at a safe distance from the fire.

“I’m at safe distance from fire”

\[ \text{BEL}_i^* \lambda x [\text{safe}(x)] \]

I thought that I was remarkably calm.

“that guy is remarkably calm”

\[ R(i, i) \land \text{BEL}_i^* \lambda x [r.\text{calm}(\gamma y[R(x, y)])] \]

\[ R = \lambda x \lambda y[\text{see.on.tv}(x, y)] \]
Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV.

“I’m at safe distance from fire”

\[ R(i, i) \land \text{BEL}_i \lambda x[\text{safe}(\gamma y[R(x, y)])] \]

\[ R = \lambda x\lambda y[x = y] \]

“I thought I was remarkably calm”

\[ R(i, i) \land \text{BEL}_i \lambda x[r\text{.calm}(\gamma y[R(x, y)])] \]

\[ R = \lambda x\lambda y[\text{see.on.tv}(x, y)] \]
Kaplan is telling the story of the time he didn’t realize his pants were on fire while seeing himself on fire on live TV

I thought I was at a safe distance from the fire

“I’m at safe distance from fire”

R(i, i) \land \text{BEL}_i \lambda x[\text{safe}(\gamma y[R(x, y)])]

R = \lambda x \lambda y[x = y]

?I thought that I was remarkably calm

“that guy is remarkably calm”

R(i, i) \land \text{BEL}_i \lambda x[\text{r.calm}(\gamma y[R(x, y)])]

R = \lambda x \lambda y[\text{see.on.tv}(x, y)]
De se names

1. Kaplan thought Kaplan was at a safe distance from the fire
De se names

1 # Kaplan thought Kaplan was at a safe distance from the fire

- Chierchia’89: Principle C blocks binding but j2-¿ coreference \(\Rightarrow\) \\
  \textit{de re} (non-\textit{de se})
De se names

1 # Kaplan thought Kaplan was at a safe distance from the fire

- Chierchia’89: Principle C blocks binding but j2-¿ coreference \(\Rightarrow\) de re (non-de se)
- cheaper alternative:

2 Kaplan thought he was at a safe distance from the fire
De se names

1 # Kaplan thought Kaplan was at a safe distance from the fire

- Chierchia’89: Principle C blocks binding but \( x^2 \) coreference \( \Rightarrow \) *de re* (non-*de se*)
- cheaper alternative:

2 Kaplan thought he was at a safe distance from the fire

- ulterior pragmatic motive for using (1)?
De se names

1 # Kaplan thought Kaplan was at a safe distance from the fire

- Chierchia’89: Principle C blocks binding but $j_2 \& \cdot$ coreference $\Rightarrow$ de re (non-de se)
- cheaper alternative:

2 Kaplan thought he was at a safe distance from the fire

- ulterior pragmatic motive for using (1)?
  - Kaplan $\in$ reported thought
De se names

1. # Kaplan thought Kaplan was at a safe distance from the fire

   - Chierchia’89: Principle C blocks binding but 2-¿ coreference \( \Rightarrow \) de re (non-de se)
   - cheaper alternative:

2. Kaplan thought he was at a safe distance from the fire

   - ulterior pragmatic motive for using (1)?
     - Kaplan \( \in \) reported thought
   - generalization: use marked coref res X only if X matches the reported thought character

# Kaplan thought Kaplan was remarkably calm

Kaplan thought the guy on TV was remarkably calm