Squib 1
LING 165C
Weiqi Yu

Degrees

1. Comparatives

Consider the following two sentences:

Sentence 1.1: Reinman is stronger than Nachenberg.[1]

Sentence 1.2: Nachenberg is stronger than Eggert.

From Sentence 1.1 and Sentence 1.2, it is entailed that Sentence 1.3:

Sentence 1.3: Reinman is stronger than Eggert.

However, our current system does not allow us to predict this, since our current semantics would only give the following things for each of them:

- $\mathsf{stronger}(\mathsf{n})(\mathsf{r})$
- $\mathsf{stronger}(\mathsf{e})(\mathsf{n})$
- $\mathsf{stronger}(\mathsf{e})(\mathsf{r})$

One might be tempted to encode the transitivity as part of $\mathsf{stronger}$ in the lexicon. However, note that every comparative such transitivity, so this property must be explained using rules that can apply to any comparative, whether we have seen or not, without changing the lexicon. Then, it might be a good idea to try to formulate a theory for the meaning of the "-er" suffix for comparatives.

At the end, for the denotation of Sentence 1.1 we would like to have something like $\mathsf{strongness}(\mathsf{r}) > \mathsf{strongness}(\mathsf{n})$, where $>$ is a transitive binary predicate on degrees. However, note that the strongness of something is not of any type that we have known: it is not an

[1]The adjective "strong" here merely refers to the physical strength (i.e., the amount of muscle) of these Computer Science professors.

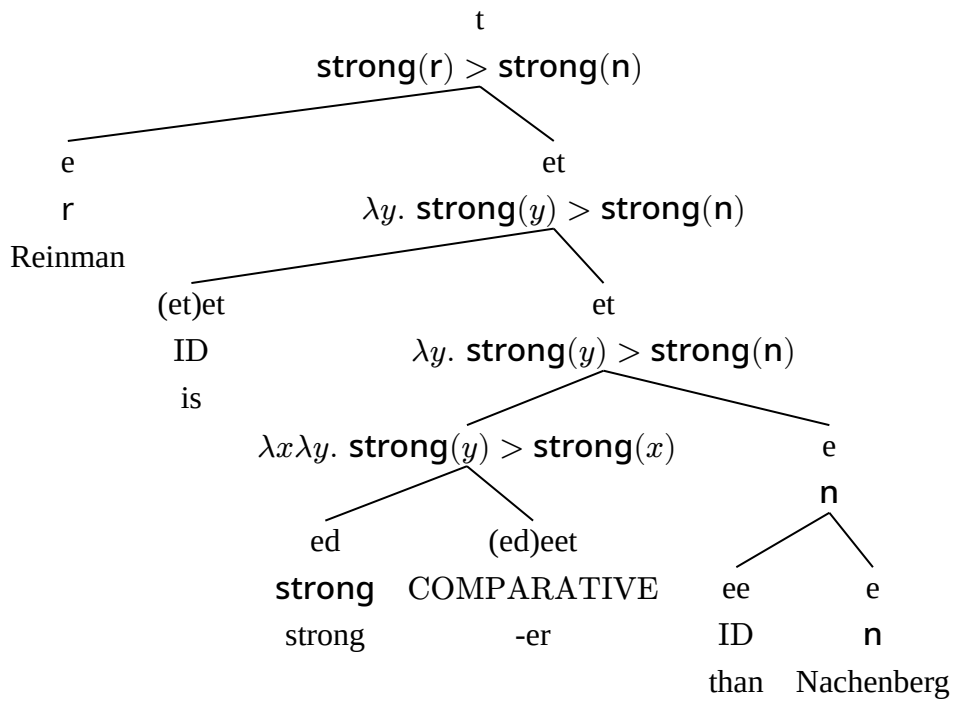
entity, not a truth value, not an event, etc. To talk about them, let us introduce a new type "d" for degrees. For example, $\mathsf{strongness}(\mathsf{r})$ would be the degree (or extent) to which r is strong.

1.1. Proposed Solution

With the degree type and our goal denotation, I propose the following semantics for adjectives and the comparative suffix "-er":

- Adjectives have the type αd, instead of the previous type αt. (α for different things to be modified like entities or events.)
- $[\![\text{-er}]\!] = \text{COMPARATIVE} = \lambda P \lambda x \lambda y . P(y) > P(x)$, where P is of type αd, and x and y are of type α. Thus, -er is of type $(\alpha d)\alpha\alpha t$.

And here is our tree for Sentence 1.1 with this semantics:



Similarly, the denotation for all three sentences would be:

- Sentence 1.1: $\mathsf{strongness}(\mathsf{r}) > \mathsf{strongness}(\mathsf{n})$
- Sentence 1.2: $\mathsf{strongness}(\mathsf{n}) > \mathsf{strongness}(\mathsf{e})$
- Sentence 1.3: $\mathsf{strongness}(\mathsf{r}) > \mathsf{strongness}(\mathsf{e})$

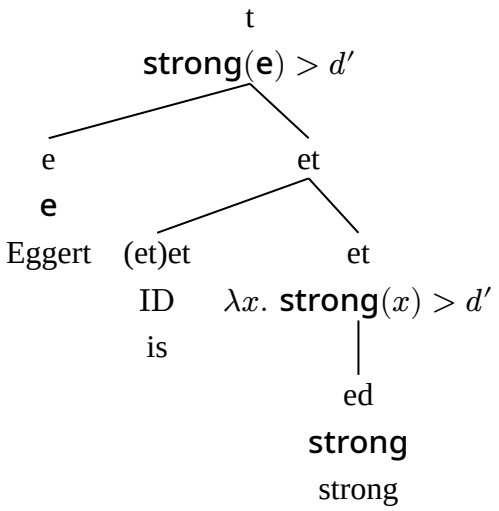
And by the transitivity of $>$, we can now predict the entailment from the conjunction of Sentence 1.1 and Sentence 1.2 to Sentence 1.3.

1.2. Problem

However, now sentences like “Eggert is strong” would be of type d, instead of t.

One way to deal with this is to introduce some fuzzy logic, so that we can claim the type of sentences not to be simply “white-or-black” binary values, but a degree of truth with “grey” ranges.

Or, if you prefer binary logic (as I do), here is the other solution: We can have some implicit type conversion (also called “coercion”) from ed to et before we compose things in such sentences:



Note that d' here is the “default” or “reference” strength known from context, such that anything that is stronger than this standard would be considered “strong”.

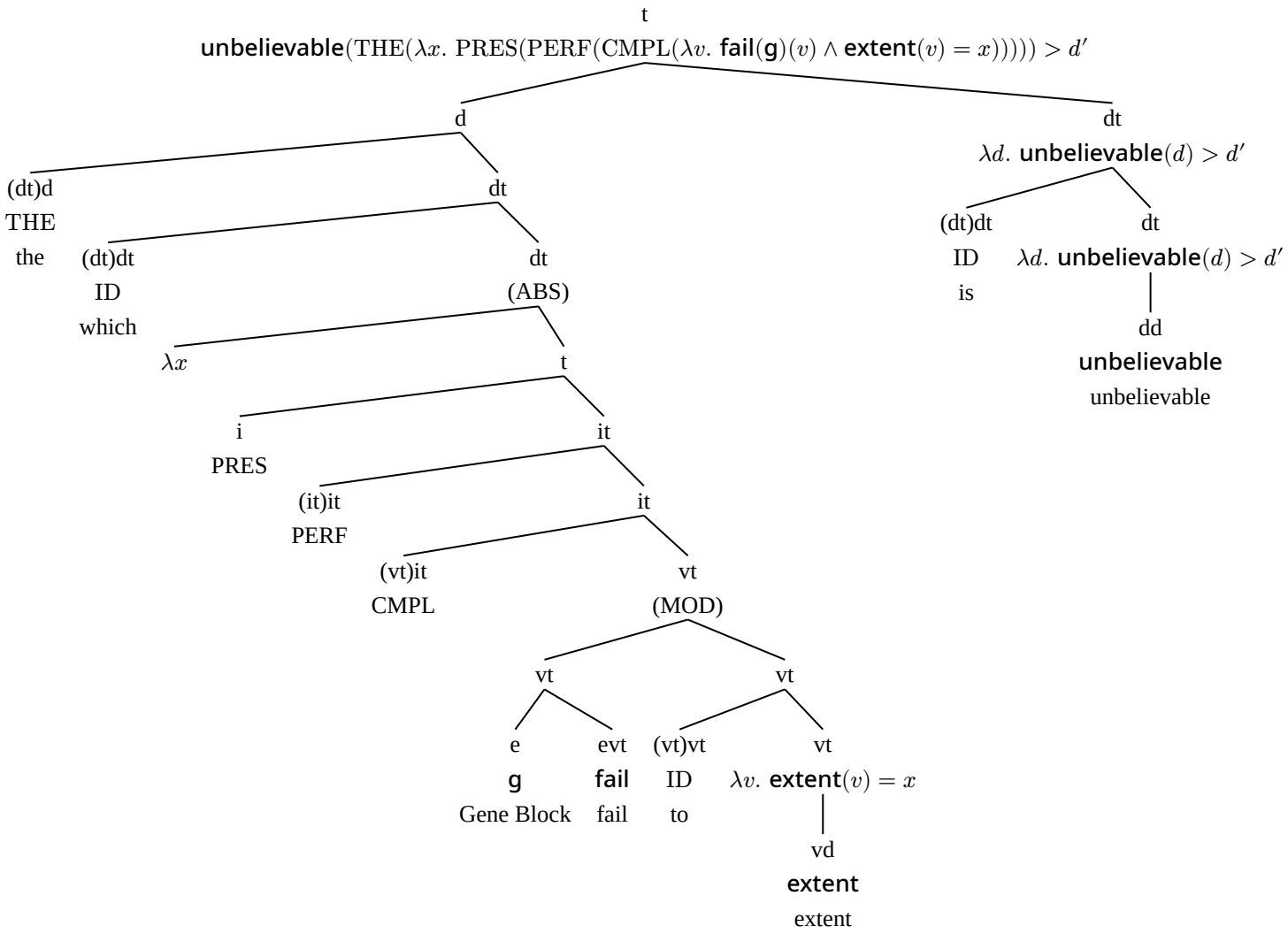
2. “extent”

So far so good. Now let us see if our theory of degrees can go futher. It turns out that now we can say about the meaning of “extent”.

Sentence 2.1: The extent to which Gene Block has failed is unbelievable.

We propose the logical form of this sentence to be as follows, where in actual speech “to” and “extent” are moved from within the VP to places above “which”:[2]

[2]Movement not drawn because I’~~m writing this paragraph an hour before the deadline and thus~~ do not have time to play around with Typst. VPISH not shown for simplicity without loss of generality. Tenses for “is unbelievable” and some intermediate denotations are not shown to save space.



The denotation for this entire sentence would be:

$$\textsf{unbelievable}(\text{THE}(\lambda x.\ \text{PRES}(\text{PERF}(\text{CMPL}(\lambda v.\ \textsf{fail}(\textsf{g})(v) \wedge \textsf{extent}(v) = x))))) > d'$$

Note that here we are generalizing "the" to be of type $(\alpha t)\alpha$, where

$$[\![\text{THE}]\!] = \lambda P.\iota x.P(x)$$

Or, in Haskell:

```haskell
-- Types of the typeclass `Dom` implement `domain` as a generic constant.
THE :: (Dom a) => (a -> Bool) -> a
THE P = _THE [ x | x <- domain, P x ]
  where _THE [x:[]] = x
        _THE _ = undefined
```

3. “Eva’s strength is the same as the kindness of Alice” / “I’m 50% moral.”

However, our system using degrees for all adjectives can be seen problematic for some adjectives that are “white-or-black”, say, “moral”: People rarely consider someone to be 50% moral, because one can only be moral or immoral. However, since the type d cannot be binary (or it would just be t again), we cannot do anything to this.

Moreover, the current typing system of our theory allows nonsensical sentences like “Eva’s strength is the same as the kindness of Alice”, since both “strength” and “kindness” return the same type d.

To solve these problems, we can introduce generics and make d not a concrete type, but a typeclass/type constraint/protocol/interface/… (whatever you want to call this). Then, we can just have different degree types for different kinds of degrees: say, strength of type en where n stands for “amount of force in Newton” (or whatever thing you want to represent strength), and kindness of type ek where k is some fancy thing that represents the amount of kindness. To be a type that satisfies the requirements of the degree typeclass, it would need to implement the function “>” and “=” (and maybe more, if more relations/operations turn out to be applicable for all degrees). We can even make type t an instance of typeclass d, so that adjectives like “moral” can stay binary, while still allowing all adjectives to be of type αd.

With generics, types of degree do not even have to be numeric – say, for sentences like “Bob is more powerful than Alice,” powerful can be of type ep where p is the degree type for power: We can make p the set of all the things one can do, and to compare between the powerfulness we can just define “>” to compare these sets in some way.