

# Transfer rules, VPM con't

---

Ling 567

March 3, 2015

# Overview

---

- Review of transfer rules
- Examples of transfer rules in detail
- Interactive development of transfer rules
- Interactive debugging of over-abundant generator output

# Anatomy of a transfer rule

- Quadruple: [CONTEXT:] INPUT [!FILTER] -> OUTPUT
- Each item above is a (partial) MRS
- Rules apply to complete MRSs to produce partially rewritten MRSs.
- Resource sensitive: INPUT is consumed in producing OUTPUT.
- CONTEXT: Additional properties beyond the INPUT that must be satisfied. (Not consumed.)
- FILTER: Negative constraints; contexts in which the rule should not apply.

# Example rule instance: pronoun insertion

---

```
pro-insert-arg1-mtr := monotonic_mtr &
[ INPUT.RELS <! !>,
  CONTEXT.RELS <! [ ARG0.SF prop-or-ques,
                    ARG1 #x & x ] !>,
  FILTER.RELS <! [ ARG0 #x ] !>,
  OUTPUT [ RELS <! [ PRED "_pronoun_n_rel",
                    ARG0 #x,
                    LBL #larg ],
          [ PRED "exist_q_rel",
            ARG0 #x,
            RSTR #harg ] !>,
          HCONS <! qeq &
                [ HARG #harg,
                  LARG #larg ] !> ],
  FLAGS.EQUAL < #x > ].
```

# Example rule instance: Pro-drop

---

```
pronoun-delete-mtr := monotonic_omtr &
  [ INPUT [ RELS <! [ PRED "pron_rel",
                      ARG0 #x,
                      LBL #larg ],
            [ PRED "exist_q_rel",
              ARG0 #x,
              RSTR #harg ] !>,
        HCONS <! qeq & [ LARG #larg,
                          HARG #harg ] !> ],
  OUTPUT [ RELS <! !>,
           HCONS <! !> ]].
```

# acm.tdl/acm.mtr

- acm.tdl is a types file
- acm.mtr is an instances file
- Transfer rules without instances won't ever do anything
- Sample acm.mtr line:

```
pro-drop := pronoun-delete-mtr
```

# Example new transfer rule: \_be+located\_v\_rel

---

```
locative_mrs_transfer_rule := monotonic_mtr &
[ INPUT [ RELS <! [ PRED "_in_p_rel",
                    LBL #lbl,
                    ARG0 #arg0,
                    ARG1 #arg1,
                    ARG2 #arg2 ] !>,
        HCONS <! !> ],
  OUTPUT [ RELS <! [ PRED "_in_p_rel",
                    LBL #larg,
                    ARG1 #arg1,
                    ARG2 #arg2 ],
          [ PRED "_be+located_v_rel",
            LBL #lbl,
            ARG0 #arg0,
            ARG1 #arg1,
            ARG2 #harg ] !>,
        HCONS <! [ HARG #harg,
                    LARG #larg ] !> ] ].
```

# Example new transfer rule: Using a transfer rule to set default values for MOOD

- In tku (and lit?) negation requires [ MOOD irrealis ], but we'd like to have [ MOOD realis ] be the default otherwise.
- Can we do this with a pair of transfer rules?
  - Idea: neg\_rel in the CONTEXT => Add [ MOOD irrealis ] to the predication qeq'ed by neg\_rel's ARG1; neg\_rel in FILTER => Add [ MOOD realis ].
- Problems:
  - With acm.tdl integrated into eng grammar, can't talk about ARG0.MOOD
  - Rule spins...



# Can we use VPM to constrain TENSE/ASPECT combinations?

- Idea:

```
E.TENSE E.ASPECT : TENSE ASPECT
present progressive <> present progressive
present * >> present *
present durative << present *
past perfective <> past perfective
past * >> past **
past perfective << past **
```

# Overview

---

- Review of transfer rules
- Examples of transfer rules in detail
- Interactive development of transfer rules
- Interactive debugging of over-abundant generator output