Is lexical insertion regulated by the subset or the superset principle?

Pavel Caha

Masaryk University (Brno)

Linguistic Flashmob
18 February 2021

- Syncretism
- Portmenteau morphemes

Latin

	war	servant
NOM	bell-**um**	serv-us
ACC	bell-**um**	serv-**um**

Latin

		war	servant
NOM	[A]	bell-**um**	serv-us
ACC	[A,B]	bell-**um**	serv-um

Latin

		war	servant
NOM	[A]	bell-**um**	serv-us
ACC	[A,B]	bell-**um**	serv-um

(1)   Lexical entries can match multiple feature structures

Latin

		war	servant
NOM	[A]	bell-**um**	serv-us
ACC	[A,B]	bell-**um**	serv-um

(1)   Lexical entries can match multiple feature structures

    a.   Subset: $um = $ [A]

Latin

		war	servant
NOM	[A]	bell-**um**	serv-us
ACC	[A,B]	bell-**um**	serv-um

(1) Lexical entries can match multiple feature structures
   a. Subset: *um* = [A]
   b. Superset: *um* = [A,B]

Korean negation (Chung 2007)

(2)   ca   -n   -ta
         sleep -pres -decl
         'is sleeping'

Korean negation (Chung 2007)

(2)   ca   -n   -ta
           sleep -pres -decl
           'is sleeping'

(3)   mos/an(i) ca   -n   -ta
           neg   sleep -pres -decl
           'can't sleep / isn't sleeping'

Korean negation (Chung 2007)

(2) ca   -n   -ta
     sleep -pres -decl
     'is sleeping'

(3) mos/an(i) ca   -n   -ta
     neg   sleep -pres -decl
     'can't sleep / isn't sleeping'
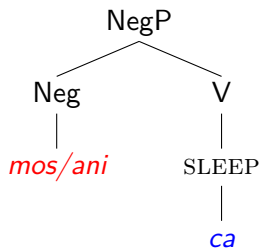
Korean negation (Chung 2007)

(4)   al   -n   -ta
        know -pres -decl
        'knows'

Korean negation (Chung 2007)

(4)   al   -n   -ta
       know -pres -decl
       'knows'

(5)   *mos/an(i) al   -n   -ta
       neg   know -pres -decl
       'cannot / does not know'

Korean negation (Chung 2007)

(4) al   -n   -ta
     know -pres -decl
     'knows'

(5) *mos/an(i) al   -n   -ta
     neg   know -pres -decl
     'cannot / does not know'

(6)   molu   -n   -ta
       neg.know   -pres -decl
     'cannot / does not know'

Korean negation (Chung 2007)

(4)   al   -n   -ta
        know -pres -decl
        'knows'

(5)   *mos/an(i) al   -n   -ta
        neg   know -pres -decl
        'cannot / does not know'

(6)   molu   -n   -ta
        neg.know   -pres -decl
        'cannot / does not know'

(7)   molu = [NEG KNOW]

Korean negation (Chung 2007)

(4)   al   -n   -ta
       know -pres -decl
       'knows'

(5)   *mos/an(i) al   -n   -ta
       neg   know -pres -decl
       'cannot / does not know'

(6)   molu   -n   -ta
       neg.know   -pres -decl
       'cannot / does not know'

(7)   molu = [NEG KNOW]

molu
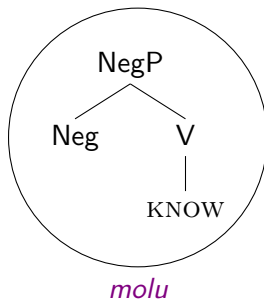
The Subset Principle

(8)   *The Subset Principle* (Halle 1997)
      The phonological exponent of a Vocabulary Item is inserted
      into a **morpheme of the terminal string** if the item
      matches all or only a subset of the grammatical features
      specified in the **terminal morpheme**.

The Subset Principle

(8) *The Subset Principle* (Halle 1997)
The phonological exponent of a Vocabulary Item is inserted
into a **morpheme of the terminal string** if the item
matches all or only a subset of the grammatical features
specified in the **terminal morpheme**.

(9) *The Modified Subset Principle* (would fail, if proposed)
The phonological exponent of a Vocabulary Item is inserted
into a **node** if the item matches all or only a subset of the
grammatical features specified in the **node**.

(10) *The Modified Subset Principle*
      The phonological exponent of a Vocabulary Item is inserted into
      a **node** if the item matches all or only a subset of the
      grammatical features specified in the **node**.
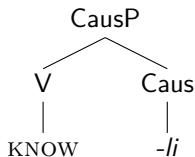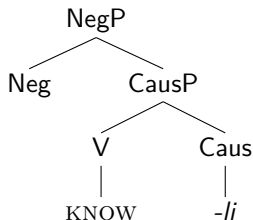
(11) *molu* ⇔ [NEG KNOW]

(10) *The Modified Subset Principle*
      The phonological exponent of a Vocabulary Item is inserted into
      a **node** if the item matches all or only a subset of the
      grammatical features specified in the **node**.

(11) *molu* ⇔ [NEG KNOW]

(12) al-li-
      know-caus
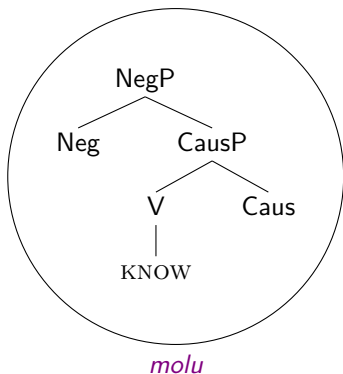   'let know, inform'

(10)   *The Modified Subset Principle*
       The phonological exponent of a Vocabulary Item is inserted into
       a **node** if the item matches all or only a subset of the
       grammatical features specified in the **node**.

(11)   *molu* ⇔ [NEG KNOW]

(12)   al-li-   CausP
       know-caus   /   \
       'let know, inform'   V   Caus
                                        | |
                                      KNOW   *-li*
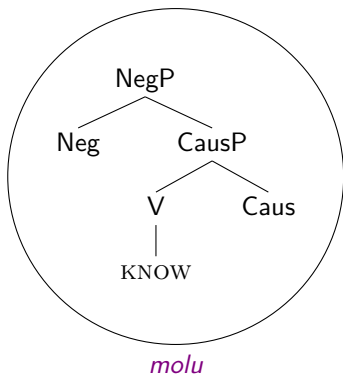
(10) *The Modified Subset Principle*
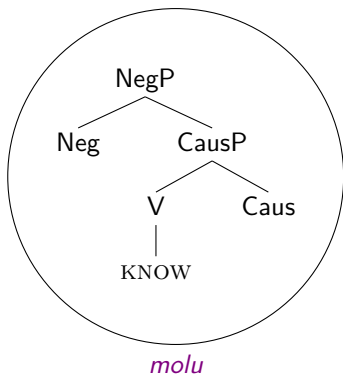   The phonological exponent of a Vocabulary Item is inserted into
      a **node** if the item matches all or only a subset of the
      grammatical features specified in the **node**.

(11) *molu* ⇔ [NEG KNOW]

(12) al-li-
      know-caus
      'let know, inform'

```
            NegP
           /    \
        Neg     CausP
               /     \
              V       Caus
              |        |
            KNOW      -li
```

(10)   *The Modified Subset Principle*
        The phonological exponent of a Vocabulary Item is inserted into
        a **node** if the item matches all or only a subset of the
        grammatical features specified in the **node**.

(11)   *molu* ⇔ [NEG KNOW]

(12)   al-li-
        know-caus
        'let know, inform'

molu

(10) *The Modified Subset Principle*
The phonological exponent of a Vocabulary Item is inserted into
a **node** if the item matches all or only a subset of the
grammatical features specified in the **node**.

(11) *molu* ⇔ [NEG KNOW]

(12) al-li-
know-caus
'let know, inform'

(13) *molu   (-li) -ess -ta
neg+know -caus -past -decl
'did not /could not inform'

molu

(10)   *The Modified Subset Principle*
       The phonological exponent of a Vocabulary Item is inserted into
       a **node** if the item matches all or only a subset of the
       grammatical features specified in the **node**.

(11)   *molu* ⇔ [NEG KNOW]

(12)   al-li-
       know-caus
       'let know, inform'

(13)   *molu   (-li) -ess -ta
       neg+know -caus -past -decl
       'did not /could not inform'

(14)   mos al-li-ess-ta
       neg know-caus-past-decl
       'did not /could not inform'

molu

Insertion

(15)   *The Superset Principle*
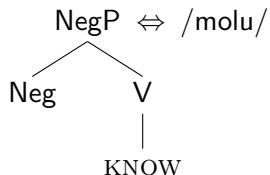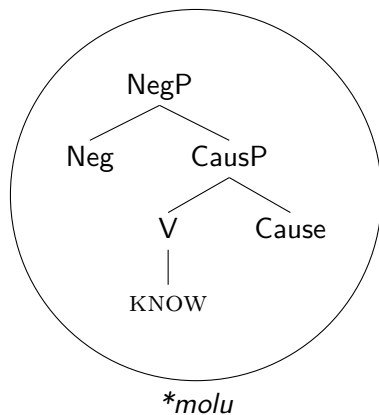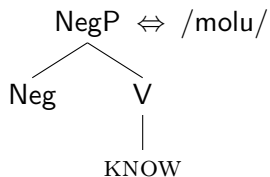   A lexically stored tree L matches a syntactic node S iff L
       contains the syntactic tree dominated by S as a subtree.

Insertion

(15)   *The Superset Principle*
A lexically stored tree L matches a syntactic node S iff L
contains the syntactic tree dominated by S as a subtree.

NegP ⇔ /molu/
Neg   V
      |
    KNOW
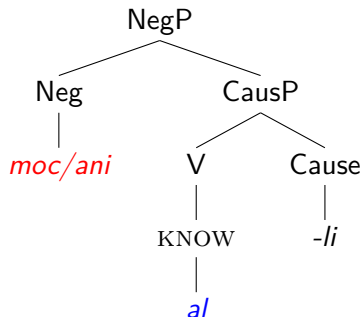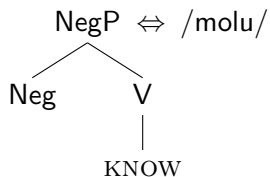
Insertion

(15)   *The Superset Principle*
A lexically stored tree L matches a syntactic node S iff L
contains the syntactic tree dominated by S as a subtree.

Insertion

(15)   *The Superset Principle*
A lexically stored tree L matches a syntactic node S iff L
contains the syntactic tree dominated by S as a subtree.

*molu

Insertion

(15) *The Superset Principle*
A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

Conclusions

- Both principles can give an account of syncretism
- The Superset Principle can also inform us when portmanteau morphemes are possible and when they are impossible

References I

Chung, Inkie. 2007. Suppletive negation in Korean and distributed morphology. *Lingua* 117. 95–148.

Halle, Morris. 1997. Impoverishment and fission. In Benjamin Bruening, Y. Kang & Martha McGinnis (eds.), *Papers at the interface*, vol. 30 MIT Working Papers in Linguistics. 425 – 449. Cambridge, Mass.