3rd Conference on Maltese Linguistics
University of Malta
08/04/2011

Indefinites and negative concord in Maltese: towards a dynamic account

Christopher Lucas
Department of Linguistics,
School of Oriental and African Studies
cl39@soas.ac.uk

1. Negative concord

In negative concord languages such as nonstandard British English (1) and Maltese (2), multiple negative expressions in a clause do not lead to multiple logical negations in interpretation, as they do in non-negative concord languages such as standard English (3).

(1) a) **Don't say *nothing* to *no one*.** (nonstandard British English)
[=standard English 'Don't say anything to anyone'.]
b) **What did you say? *Nothing!***
[same interpretation as in standard English]

(2) a) **It-tifla ma rat *xejn*.** (Maltese)
the-girl neg see.prf.3fs n.thing
'The girl didn't see anything.'
b) **X'rat? *Xejn!***
what-see.prf.3fs n.thing
'What did she see?' 'Nothing!'

(3) ***No one* said *nothing*.** (standard English)
[double logical negation (¬¬p=p) in interpretation]

2. N-words

(4) An expression α is an n-word iff:
(a) α can be used in structures containing predicate negation or another α-expression yielding an interpretation of the clause as containing only one logical negation; and
(b) non-sentential/elliptical strings containing α and lacking a negator can be interpreted negatively.
(cf. Giannakidou 2005: 328)

A hard question:

- Are n-words negative?
  - No consensus within mainstream theories

A Dynamic Syntax approach to negative concord offers an alternative by formalizing two intuitions:

a) Negation is a property of sentences/propositions rather than lexical items per se.
b) An n-word's contribution to the negativity of a sentence is **context-dependent**.

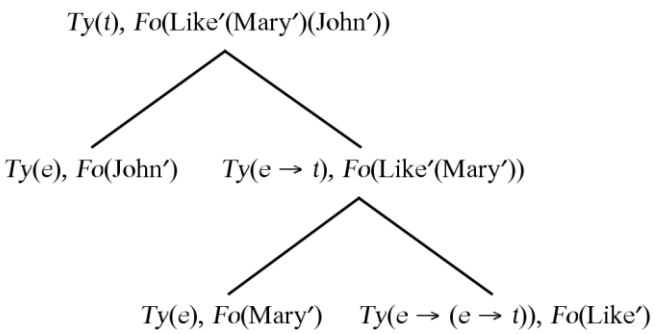
3. Dynamic Syntax

Dynamic Syntax (Kempson et al. 2001, Cann et al. 2005) is a grammar formalism that models the real-time, incremental construction of semantic representations of content from strings of words uttered in context.

- Predicate-argument structure is the only level of representation.
- This is depicted on trees, which therefore represent *the interpretation of words in context*, **not** the constituent structure of words in a string.

An example of a basic tree structure:

(5) **John likes Mary.**

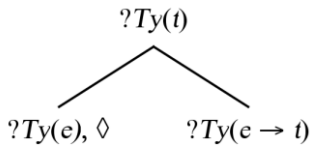


- Trees are gradually constructed and 'decorated' with semantic information (primarily type (*Ty*) and formula (*Fo*) values) through a combination of general transition rules and 'lexical actions'.
- Lexical actions are instructions to (among other things) decorate nodes and build new tree structure. These actions are carried out as words are parsed on a left-to-right basis.

- A key feature of this incremental system is underspecification (of content, structure, structural relations), together with requirements for specification in context.
- Requirements are expressed as decorations on nodes and typically pertain to the establishment of particular formula values.
- The symbol for a requirement is a question mark: **?**

The initial state of a tree is, roughly, as in (6), with the root node decorated with a requirement to establish a propositional formula, and an argument daughter with a $?Ty(e)$ requirement (subject) and a functor daughter with a $?Ty(e \rightarrow t)$ requirement (predicate):

(6)



The 'pointer' ◊ indicates the currently active node in the tree.

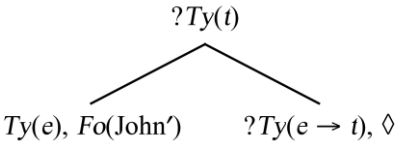
The tree is gradually decorated (and potentially grows new nodes) as words are parsed, their lexical entries are accessed, and the specified lexical actions are carried out, as in (7).

(7) *John*

IF	$?Ty(e)$	Trigger
THEN	`put`($Ty(e)$, Fo(John′))	
ELSE	`Abort`	Failure

Since the pointer in (6) is at a node with a $?Ty(e)$ requirement, the trigger in (7) is fulfilled and the action to decorate the current node with $Ty(e)$ and Fo(John′) is carried out. The requirement at this node is thus fulfilled, and the pointer can move to the sister node, giving us the partial tree in (8).

(8) **John...**



As each word in the string *John likes Mary* is parsed, the various tree nodes are built and/or annotated with type and formula information, until finally the ?*Ty*(*t*) requirement at the root node is satisfied and a propositional formula is established, as in (5).

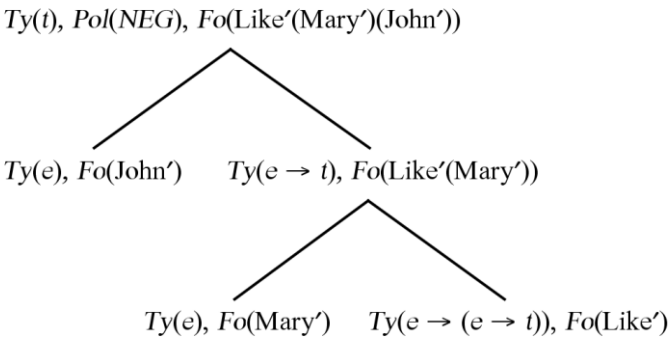
- A string of a given language is well-formed iff there is at least one possible sequence of actions that results in a tree with no outstanding requirements.

4. Negation in Dynamic Syntax

Recent developments in Dynamic Syntax make use of a more intricate structure than that presented in (5), whereby every predication has an extra situation/event argument which is central to accounts of tense, aspect, mood etc. (which will be ignored here).

A full account of negation would make use of the situation argument, but for simplicity we can represent negation as a feature *Pol*(*NEG*) (i.e. a negative value for the polarity of the proposition) decorating the ?*Ty*(*t*) node, as in (9), which is otherwise identical to (5).

(9) **John doesn't like Mary.**



Key difference between n-words and straightforward negative expressions:

- True negative expressions always automatically decorate the ?*Ty*(*t*) node with *Pol*(*NEG*).
- N-words are sensitive to whether the ?*Ty*(*t*) node already has this decoration or not.

4.1 True negative expressions

True negative expressions satisfy the second clause of (4) – they are interpreted negatively on their own in nonsentential utterances – but not the first: when they combine with predicate negation or another true negative expression, the result is more than one logical negation in interpretation:

(10) John doesn't like *nothing*. (standard English)

We can capture this behaviour with the type of lexical entry in (11).[1]

(11) *nothing*

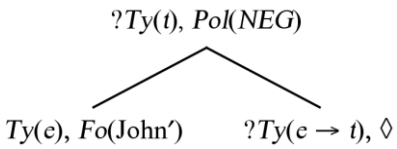
IF	$?Ty(e)$
THEN	`put`($Ty(e)$, $Fo(\varepsilon, x, \text{Thing}'(x))$);
	`gofirst`($?Ty(t)$);
	`put`($Pol(NEG)$)
ELSE	`Abort`

Predicate negators such as *not* will have a very similar lexical entry in these languages:

(12) *not*

IF	$?Ty(e \rightarrow t)$
THEN	`gofirst`($?Ty(t)$);
	`put`($Pol(NEG)$)
ELSE	`Abort`

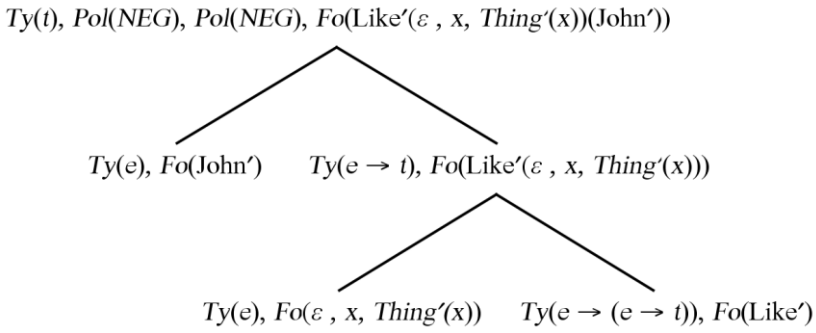
(13) **John doesn't...**



[1] Quantification in DS, which we do not present in detail here, is expressed in terms of Hilbert and Bernay's (1939) epsilon calculus, where quantified expressions are treated as naming an arbitrary witness of the set denoted by the restrictor. See Kempson et al. (2001: ch.7) for details.

Completed tree for *John doesn't like nothing* with the lexical entry in (11) for *nothing*:

(14) **John doesn't like nothing.**



3.2 Negative concord in Maltese

So-called 'strict' negative concord languages like Maltese (or Slavic, Romanian, etc.) have the following characteristics:

- When n-words co-occur with each other and/or with predicate negation, there is only one logical negation in interpretation.
- Every (non-elliptical) negative sentence must have negation marked on the predicate.
- All indefinite pronouns in the scope of negation must be n-words.

Additional characteristics of Maltese negation:

- Negative sentences without indefinite pronouns must have the bipartite construction *ma...-x* (just one of the two elements is insufficient), **BUT**...
- Negative imperatives generally lack a preverbal negator, **AND**...
- N-words and *-x* do not co-occur in a clause with a single logical negation in interpretation.

Data (from Haspelmath and Caruana 1996):

(15) It-tifla ma rat-*(x) lil-tifel.
the-girl neg see.prf.3fs-neg to-the-boy
'The girl didn't see the boy.'

(16) It-tifla ma rat *xejn* (/**xi ħaġa*).
the-girl neg see.prf.3fs n.thing (/anything)
'The girl didn't see anything.'

(17) *Hadd* *(ma) qal-li(*-x) *xejn*.
n.body neg say.prf.3ms-to.me n.thing
'Nobody told me anything.'

(18) a) X'rat? b) *Xeyn*.
what-see.prf.3fs n.thing
'What did she see?' 'Nothing.'

Some generalizations over (15)–(18):

i) *-x* is always associated with a negative interpretation[2] and at least some sentences with *-x* alone are well-formed.
ii) *Ma* is always associated with a negative interpretation but a sentence with (negative) *ma* alone is never well-formed.
iii) *Xeyn/ħadd* etc. are always associated with a negative interpretation, but when they combine there is only a single logical negation in interpretation.

In Dynamic Syntax terms:

i) *-x* always automatically decorates the $?Ty(t)$ node with $Pol(NEG)$.
ii) *Ma* decorates the $?Ty(t)$ node with $?Pol(NEG)$, i.e. a **requirement** for $Pol(NEG)$, not the feature itself.
iii) *Xeyn/ħadd* etc. decorate the $?Ty(t)$ node with $Pol(NEG)$ **if and only if** it does not already have this decoration.

Lexical entries:

(19) *-x*

IF	$?Ty(e \rightarrow t)$	
THEN	`gofirst`$(?Ty(t))$	
	IF	$\exists \mathbf{x}.Pol(NEG)(\mathbf{x})$
	THEN	`Abort`
	ELSE	`put`$(Pol(NEG))$
ELSE	`Abort`	

This is similar to the lexical entry for English *not* in (12) except that (19) aborts the derivation if there has already been a $Pol(NEG)$ decoration at the moment *-x* is parsed, cf. (17).

[2] I assume that the question-marking particle *-x* is a separate lexical item to negative *-x*.

(20) *ma*

IF	$?Ty(e \rightarrow t)$
THEN	`gofirst`$(?Ty(t))$;
	`put`$(?Pol(NEG))$
ELSE	`Abort`

Here the $?Pol(NEG)$ requirement ensures that there must some other expression in the sentence which can make a $Pol(NEG)$ decoration, since a tree with any unsatisfied requirements is ill-formed, cf. (15).

(21) *xeyn*

IF	$?Ty(e)$	
THEN	`put`$(Ty(e), Fo(\varepsilon, x, \text{Thing}'(x)))$;	
	`gofirst`$(?Ty(t))$	
	IF	$\neg\exists \mathbf{x}.Pol(NEG)(\mathbf{x})$
	THEN	`put`$(Pol(NEG))$
	ELSE	`go`$(\langle\downarrow_*\rangle ?\varphi)$
ELSE	`Abort`	

This is similar to the lexical entry for English *nothing*, except here the $Pol(NEG)$ decoration only takes place if there has not already been such a decoration, cf. (16)–(18).

This is the defining property of all n-words in Maltese (presented in Table 1). Non-negative indefinites such as *xi ħaġa* 'something/anything', by contrast, `Abort` if they are parsed after some other element has already made a $Pol(NEG)$ decoration, cf. (22).[3]

Table 1: Maltese indefinites (after Haspelmath and Caruana 1996)

	n-words	non-negative
Determiner	ebda	xi
Thing	xejn	xi ħaġa
Person	ħadd	xi ħadd
Time	qatt	xi darba
Place	imkien	xi mkien

[3] This is a simplification, ignoring issues of relative scope. *Xi*-series indefinites are not ungrammatical following negation, but they obligatorily take high scope, e.g. *Jekk ma ħimtx xi ħaġa...* : if > thing > neg > understand thing (*If > neg > understand > thing).

(22) *xi ħaġa*

```
IF      ?Ty(e)
THEN    put(Ty(e), Fo(ε, x, Thing′(x)));
        gofirst(?Ty(t))
        IF      ∃x.Pol(NEG)(x)
        THEN    Abort
        ELSE    go(⟨↓*⟩?φ)
ELSE    Abort
```

Obligatoriness of *ma* despite the presence of other negative elements (except in imperatives) is expressed as a property of affirmative indicative verbal morphology:

(23) *AFFIRMATIVE INDICATIVE MORPHOLOGY*

```
IF      ?Ty(e → t)
THEN    gofirst(?Ty(t));
        put(Pol(AFF))
ELSE    Abort
```

The *Pol(AFF)* decoration (i.e. an affirmative/veridical value for the polarity of the proposition) in (23) ensures *ma* is obligatory in any sentence containing some other negative element, since conflicting *Pol*(AFF) and *Pol*(NEG) decorations will render a derivation ill-formed, cf. (17) and the ungrammatical (24), in which an n-word follows a verbal predicate with no *ma* prefix.

(24) *It-tifla rat *xejn*
the-girl see.prf.3fs n.thing
Intended: 'The girl didn't see anything.'

3.3 The problem of qatt

In some strict negative concord languages (e.g. Catalan, which is 'optionally strict') n-words may appear in nonveridical contexts other than negation (e.g. interrogatives, conditionals) with non-negative meaning.

Maltese seems to have one such n-word, *qatt* '(n)ever':

(25) Ma niċħadek *qatt*.
neg deny.impf.1s-2s (n)ever
'I will never deny you.'

(26) Jekk *qatt* tiġi Londra, ejja arani.
if (n)ever come.impf.2s London come.imp.2s see.imp.2s-me
'If you ever come to London, come and see me.'

This can be captured by assuming that all nonveridical operators decorate the $?Ty(t)$ node with an appropriate 'force' feature $Force(NV)$ (which is entailed by more specific decorations such as $Force(INT)$ or $Pol(NEG)$). Whether *qatt* decorates the $?Ty(t)$ node with $Pol(NEG)$ is then sensitive to whether it is already has a $Force(NV)$ decoration:

(27) $qatt_{NEG\_CONTENT}$

IF	$?Ty(e)$	
THEN	`gofirst`$(?Ty(t))$	
	IF	$\neg\exists \mathbf{x}.Force(NV)(\mathbf{x})$
	THEN	`put`$(Pol(NEG))$
	ELSE	`go`$(\langle\downarrow_*\rangle ?\varphi)$
ELSE	`Abort`	

Thus *qatt* will only cause a sentence (or sentence fragment) to be negative if no other element has already made it nonveridical.

3.4 Unresolved issue

The lexical entry in (19) rules out (28) but not (29).

(28) **Xejn* ma waqa-*x*.
n.thing neg fall.prf.3ms-neg
Intended: 'Nothing fell.'

(29) *It-tifla ma rat-*x* *xejn*
the-girl neg see.prf.3fs-neg n.thing
Intended: 'The girl didn't see anything.'

1. Simple (but unsatisfactory) excuse:

Maltese is exceptional compared to similar Arabic dialects in having:

a) (apparently?) strict mutual exclusivity between *-x* and n-words
b) no possibility of marking negation with *ma* alone
c) some contexts in which *-x* alone can mark negation (i.e. imperatives)

Other Arabic varieties with *ma…-š* negation generally either have (a) but not (b) or (c) (Maghrebian dialects), or (b) and (c) but not (a) (Palestinian and Egyptian).

Maltese having all three properties is thus synchronically odd and presumably diachronically unstable.

2. Hint at a more satisfactory (but more complex) solution:

A fuller Dynamic Syntax account of Maltese negation (involving the situation argument ignored here) would have *-x* make its *Pol*(*NEG*) annotation on a different (lower) node to that of n-words.

N-words would then involve three different possible actions:

a) make a *Pol*(*NEG*) annotation if no other n-word has already
b) make no *Pol*(*NEG*) annotation if another n-word has already
c) `Abort` if *-x* has already made its *Pol*(*NEG*) annotation

This kind of distinction is independently motivated by the fact that in biclausal structures the negation associated with *-x* never seems to be interpretable in a higher clause than the one it appears in (30), whereas the negation of n-words routinely is (31).

(30) Ordnalu ma jiċċaqlaqx.
order.prf.3ms-to-him neg move.impf.3ms-neg
'He ordered him not to move.'
[Not possible: 'He didn't order him to move.']
(Borg and Azzopardi-Alexander 1996: 93)

(31) M'għandi aptit nagħmel xejn.
neg-have-1s appetite do.impf.1s n.thing
'I don't feel like doing anything.'

4. Conclusion

General message:

The context-sensitivity of lexical entries and the incrementality of interpretation in Dynamic Syntax provides a solid basis for an account of meaning in natural language that is strictly compositional throughout, without the need to posit

invisible elements, large-scale homophony etc., or to consign a vast range of constructions to a non-compositional periphery.

Specific message:

The apparently unstable behaviour of n-words in Maltese and other languages, as well as the different types of negative concord in general, are hard problems to account for in strictly compositional fashion in traditional formalisms. By contrast, the context-sensitivity at the heart of Dynamic Syntax is ideally suited to providing such an account.

References

Borg, Alexander, and Azzopardi-Alexander, Marie. 1996. *Maltese*. London: Routledge.

Cann, Ronnie, Kempson, Ruth, and Marten, Lutz. 2005. *The Dynamics of Language: An Introduction*. Amsterdam: Elsevier.

Giannakidou, Anastasia. 2005. 'N-words and negative concord'. In Martin Everaert, Henk Van Riemsdijk, Rob Goedemans and Bart Hollebrandse (eds.), *The Blackwell Companion to Syntax*. Oxford: Blackwell.

Haspelmath, Martin, and Caruana, Josephine. 1996. 'Indefinite pronouns in Maltese'. *Rivista di Linguistica* 8: 213–227.

Hilbert, David, and Bernays, Paul. 1939. *Grundlagen der Mathematik II*. Berlin: Springer.

Kempson, Ruth, Meyer-Viol, Wilfried, and Gabbay, Dov. 2001. *Dynamic Syntax: The Flow of Language Understanding*. Oxford: Blackwell.