4 Computational implementation of crossed control structures in Indonesian

ARKA, I Wayan
Australian National University/Fakultas Sastra dan Budaya, Universitas Udayana
wayan.arka@anu.edu.au

Abstract

A crossed/backward control structure in Indonesian is analysed within LFG as forming a complex SVC-like tight structure that allows argument sharing. It shows properties having intermediate status between a complex predicate and an ordinary complementation structure. The analysis is implemented and tested in the computational grammar of Indonesian, developed with the ParGram (Parallel Grammar) framework.

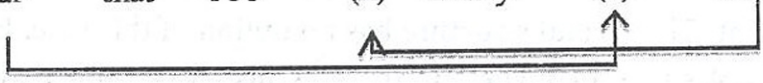
Keywords: *backward/crossed control, Serial Verb Construction (SVC), Lexical Functional Grammar (LFG), Computational Grammar, Parallel Grammar.*

1 Introduction†

This paper reports the computational implementation of the so-called backward or crossed control structures in Indonesian, exemplified in ((1)). In this example, there are two control relations: the forward control relation associated with the fronted P(atient) argument mobil itu and the backward control relation associated with the A(gent) argument *ku=*. The forward and backwards control relations are indicated by the two arrows. The control paths cross with each other, and the structure with these control relations is therefore often called a structure with crossed control.

(1)

Mobil	*itu*	*yang*	__	*coba*	[__	*ku=jual*].
car	that	FOC	(A)	UV.try	(P)	1SG=UV.sell



'That car (is the one that) I tried to sell.'

One of the salient and interesting properties of control verbs that allow the crossed control reading is that they can give rise to puzzling ambiguity as seen in ((2)). In the first reading, which is an ordinary control structure, the matrix subject *anak itu* is the experiencer and the controller of the embedded subject. In the second reading, the matrix subject is only the controller of the embedded subject but is not the experiencer. The experiencer shows up in the embedded clause as an oblique (*oleh ibu*).

(2)

Anak itu	*mau/ingin*	[–	*di-cium*	*oleh*	*Ibu*]
child that	want		PASS-kiss	by	Mother

† This paper was originally a section of the paper presented at the LFG2014 at Ann Arbor, Michigan, July 17-20, 2014. Special thanks go to audiences at the conference, in particular Paul Kroeger, Ron Kaplan and Mary Dalrymple, for their feedback. Research reported here was made possible because of the financial support from the ARC grant (DP0877595, 2008-2011). And the write-up of the paper was mainly done in Germany with generous support from the Humboldt Foundation, who provided me with research fellowship (2012-2014).

i. 'The child wanted to be kissed by Mother.' (ORDINARY CONTROL READING)
ii. 'Mother wanted to kiss the child.' (BACKWARD/CROSSED CONTROL READING)

There is a complex interplay between syntax, semantics and pragmatics in control structures that allow the crossed control reading. For Indonesian/Malay, the issue of crossed control has been discussed in different frameworks, e.g. Polinsky & Potsdam (2008) discuss it in relation to the verb *ingin/mau* and Nomoto (2011) discusses it in relation to other control verbs.‡ Both are within the Chomskyan framework. Building on these works, Arka (2014) develops an analysis within LFG (Lexical-Functional Grammar) (Bresnan 2001, Dalrymple 2001, Falk 2001), addressing certain issues that appear to be problematic in the previous analyses.

The computational implementation reported in this paper is based on the LFG analysis as discussed in Arka (2014).

The paper is organised as follows. In section 2, the basic points of the linguistic analysis in Arka (2014) are presented. In section 3, an overview of the computational Indonesian ParGram is given, followed by the computational implementation of the analysis of crossed control. Section 4 outlines future work within ParGram research.

2 Crossed control: a highly constrained control structure

The following are the key points of Arka's (2014) linguistic analysis of control and crossed control structures in Indonesian.

Firstly, there are two available structures that a control verb like *mau/ingin* 'want' and *coba* 'try' can enter: the ordinary complementation structure and the tight SVC-like structure. The two in LFG format can be represented in ((3)). As seen in ((3)a), the complementation structure is not so tight. There is a possibility for material intervention (XP) between the first verb, which is filled in by a control verb, and the second VP complement. The second structure has no option of this type. Evidence for this comes from examples like ((4)) where in the presence of the intervening material (e.g. *agar*), backward reading is blocked.

(3) VP → a. V (XP) VP (complementation-VP)
↑=↓ (↑XCOMP)=↓

b. VP VP (SVC-VP)
↑=↓ (↑XCOMP)=↓

(4) *Anak itu ingin [agar __ di-cium oleh Ibu]*
= child that want AGAR PASS-kiss by Mother
i. a) 'The child wanted to be kissed by Mother.' (FORWARD CONTROL ONLY)
or, b) 'The child wanted that s/he would be kissed by Mother.'
ii. * 'Mother wanted to kiss the child.' (BACKWARD CONTROL)

Secondly, the tight VP structure in ((3)b) has properties that resemble a complex predicate, even though it is perhaps not a truly complex predicate. A real complex predicate is monoclausal§, and Indonesian does have a V-V complex predicate, *putar balik* as in ((5)), quoted from Tempo online. Both *putar* 'turn around' and *balik* 'turn

‡ Other work discusses crossed control includes Arka (2000), Gil (2002), and Sato and Kitada (2012)
§ Davies (2014) argues that crossed control structures in Madurese are indeed monoclausal.

upside down' are transitive verbs with similar meanings, and they can appear by themselves as clausal predicates. As seen in ((5)), the two can form a complex predicate, treated as a single a complex stem taking the prefix *meN-* and suffix *–kan.*

(5) *Aparat* *keamanan* *di-nilai* *sering*
authority security PASS- consider often

memutar-balik-kan *fakta* *atas* *setiap* *insiden ...*[**]
AV.spin-upside.down-CAUS fact P every incident
'The security authority has been often regarded as twisting the facts of any incident ...'

In addition to affixation, evidence from possessive reflexive *sendiri* 'own' also provides good evidence for monoclausality. Possessive reflexive *sendiri* can only have its antecedent within its own predicate nucleus. First consider the example of possessive reflexive *sendiri* without a complex predicate:

(6) *John* *menyuruh* *saya* *memakai* *mobil* *sendiri.*
John AV.ask 1SG AV.use car self.
'John asked me to use my own/*his car.'

Sendiri in the example above cannot refer to *John* because *John* is not an argument of the predicate *memakai,* which serves as the nucleus of the object NP (*mobil sendiri*). With *putar balik* and also *jungkir balik, sendiri* can be bound by the subject NP as seen in (7a-b).

(7) a. *John* *memutar-balik-kan* *fakta* *sendiri*
John AV.spin-upside.down-CAUS fact self
'John twisted his own facts.'

b. *John* *men-jungkir-balik-kan* *mobil* *sendiri.*
John AV.turn.around-upside.down-CAUS car self
'John turned his own car upside down.'

The backward control structure with possessive reflexive, however, shows that the two verbs do not form a (monoclausal) complex predicate. Consider the following sentence where the P with *sendiri* in the subject NP can only be bound by *kita* (its co-argument within the same embedded/second predicate nucleus *bawa*), not by *dia* (the matrix A argument).

(8) *Apakah* *mobil* *sendiri* $_{j/*i}$ *yang* *dia* $_{i}$ *suruh*
QW car self REL 3SG UV.ask

kita $_{j}$ *bawa?*
1PL.Inc UV.take
'Did he ask us to drive our own car/*his (own) car?'

The inability of A (*dia* '3SG') to function as the antecedent of the possessive reflexive *sendiri* in ((8)) suggests that the first verb *suruh* and the second verb *bawa* do not form a single nucleus. In other words, the structure with crossed control is not monoclausal.

However, while not monoclausal, the VP that allows backward/crossed control (labelled as the SVC-VP here) is very tight, and different from a normal VP

[**] http://www.tempo.co/read/news/2012/01/24/063379232/Aparat-Dituding-Memutar-Balik-Fakta-Mesuji

complement structure. Evidence for this, as pointed out earlier, comes from certain properties like no intervention (cf. example ((4))).

In short, there is good evidence that the SVC VP is structurally intermediate between a true complex predicate and an ordinary complement VP. Like a complex predicate, the structure is tight enough to allow argument sharing and to be expressible only once. Like an ordinary complementation structure, it still shows some bi-clausal properties as seen from reflexive binding. For these reasons, the adopted SVC-VP analysis takes into account both VP properties (e.g. the verbal complex is a string of VPs with its V(P) component showing some independence in terms of, say, having its own non-shared argument) and complex predicate properties (e.g. the whole top-most VP unit has a distinct subcategorisation frame that is not the same as either the first or the second VP). In other words, we have an SVC construction, composed by the VP units. The idea of predicate composition is important and is implementable computationally, as discussed in the next subsection.

3 Computational implementation in Indonesian ParGram

This sub-section outlines the computational implementation and testing of the analysis of Indonesian crossed control in the XLE-based Indonesian Parallel Grammar (IndonGram).[††] Our IndonGram currently can handle the different voice types and syntactic structures in Indonesian, including applicative and causative alternations (Arka et al. 2009). Each voice type comes with a template (not discussed here, but see (Arka et al. 2009)) and imposes changes in the input subcategorisation frame, formulated as classic lexical rules in LFG (Bresnan 1982).

We have enhanced the capability of IndonGram to be able to handle crossed control. As discussed in the preceding section, this kind of control requires a tight SVC-like structure. Treating the tight SVC-structure as a complex predicate allows us to capture argument raising/sharing, enabling an argument to be realised only once in the surface syntax. The implementation in XLE (Crouch et al. 2007) makes use of the restriction operator (Kaplan and Wedekind 1993, Butt, King, and Maxwell III 2003). The relevant components for the implementation include the annotated c-str, the SVC template capturing predicate composition and argument sharing, and the lexical entry specification.

The relevant c-str showing the VP for this, generated by the phrase structure rule shown in ((3)), is given in ((9)a). As seen, the c-str comes with the necessary annotation, including its SVC template (@SVC). Sample entries for control verbs such as *mau*, *minta*, and *coba* are given in ((10)).

[††] https://pargram.b.uib.no/