

Negotiation - Final Coding Manual

Elijah Mayfield

October 5, 2010

1 Defining Four Core Moves

There are two axes of information exchange in a traditional sequence. While the exact sequence will be defined below, we must first look at these two interacting dimensions of utterances. The first element of these negotiation is whether the topic is knowledge (**K**) or action (**A**). The second element is whether a speaker is presenting themselves as a primary (**1**) or secondary (**2**) authority for the topic at hand.

What does this mean in practice? Four codes can be defined through these two dimensions: giving knowledge on which you consider yourself the primary authority (**K1**), asking for knowledge from someone who you consider a primary authority over yourself (**K2**), showing (through narration) that you have the authority to perform an action (**A1**), and instructing others to perform an action, which gives them the primary authority to perform it or reject your command (**A2**). Any move which does not fit into these sequences should be labelled as other (**o**). The decision tree for these labels is given in Figure 2.

1.1 Clarifications

- *Authorizing* the action of a group you are in counts as an **A1** action. *Suggesting* the action of a group you are in counts as an **A2** action. *Narrating* an action you are performing on behalf of a group you are in counts as an **A1** action. *Requesting* action from someone counts as an **A2** action.

student2 A2 Let's buy two bowls. (*suggesting to group*)
student3 A1 Okay. (*authorizing on behalf of group*)
student2 A2 Can we get two bowls? (*requesting action from a facilitator*)

student2 A1 Let me see how this works. (*narrating own actions*)

All other core moves should be classified as **K** moves.

- In addition to delaying moves, there are also moves which track the information in a previous utterance without actually serving as a K2 move. For instance:

student3 K2 Is this uh, transparency sheet
student2 K1 That is just money.
student3 o Oh that's money?
student2 o I think so

This repeated question is just tracking the statement made by the previous sequence, not adding new information, so they can both be labelled as **o** moves.

2 Defining a Martin Sequence

While these codes are useful, they are not helpful if they are viewed in isolation. It is only through an understanding of the interaction between speakers in a discourse that these codes can be interpreted meaningfully. To give ourselves a framework in which to interpret “context” we turn to more insights from the tradition of Systemic Functional Linguistics.

Martin’s theory of Discourse declares that a standard interaction between two speakers will match a specified pattern exactly. This pattern is:

o* X2? o* X1 o*

This is to say that there is a single key move in an exchange between speakers, at which a new proposition is introduced by someone with the authority to make that introduction. This can correspond to a **K1** or **A1** move. Leading up to this, there may be delaying moves, backchannel moves, false starts, or other non-contentful moves. They could be attention-grabbers to gain the floor in a multi-party interaction or they could be any other move which does not introduce new primary knowledge into the discourse. These are the *o* moves in the pattern above.

A primary authority utterance may be a response to a request in the form of a secondary authority move. That is the place of the **X2** move in the formula above. In practice, this is the same surface structure as an adjacency pair from the field of conversation analysis. Typical interactions of this form may include a question/answer pair (**K2-K1**). or a command-compliance pair (**A2-A1**).

2.1 Clarifications

- A move which initiates one of these sequences does not have to be a core move.

	(original Martin)	(simplified)	
student2	dK1	o	Can I give you a hint?
student1	dK2	o	Sure.
student2	dK1	o	What equations can you use here with these variables?
student1	K2	K2	The quadratic equation.
student2	K1	K1	Right.

This sequence is one continuous buildup to the **X1** move from Martin’s pattern. The first statement is a delaying action, getting permission to take the floor in an exchange and perform an illocutionary act. The second line is again a preparatory and delaying move, giving authority to the first speaker for him to perform the action. The third line is phrased as a **K2** move, but we can see in the context that the speaker has full authority over the answer. That is what makes the penultimate move a **K2** move even though it is phrased as a statement. It is requesting the approval of the other student, which is given in the last line (a **K1**).

- There is no well-defined length for how long a sequence must be, in terms of number of utterances. In fact, there may be times in which multiple new sequences are all started in rapid succession, each sequence lasting no more than one or two utterances. This is especially true if the sequences are being broken (see below). On the other hand, a series of utterances all in a row may make up a single complex move, such as a **K1** move extending over several sentences as the new content is elaborated on or restated.
- Sometimes, multiple moves in a row will all form part of the same core move. For instance, elaboration might occur over multiple continuous lines from the same speaker, or the same proposition might be stated and then restated. These can be treated as a single move, and each can be labelled as a **K1** within the same thread.

- With that being said, whenever another speaker enters into a series of moves, a complex is broken and must be treated as separate moves. When multiple speakers are introducing new information in a series of **K1** moves, each move is starting a new sequence, unless it is directly referencing or restating a statement from a previous sequence, in which it may be an **o** move from that prior sequence.

3 Breaking a Martin Sequence

This theory of discourse is not meant to imply that all interactions between speakers match this pattern exactly in all circumstances. On the contrary, this is only the simplest case. Conversation is multifaceted and speakers switch between topics rapidly. Some sequences may be started but not picked up on by the other speakers in an interaction. There are two distinct cases where a new utterance may not fit into the pattern of an unfinished sequence.

The first is related to the concept of threadedness in discourse, when more than one topic can be discussed simultaneously. This results in dropping the old sequence before it reaches the compulsory **X1** move and opening a new thread of conversation while leaving the previous one available for future use. This can happen either by introduction of a new sequence or returning to a previous sequence. This type of return, shifting conversation back to a topic that had already been introduced but which had not been resolved satisfactorily, can also happen due to a variety of reasons and can be expressed in many different ways in conversation.

The other type of sequence break is a challenge move, in which the authority of the previous speaker is being questioned, meaning that the current speaker thinks the previous speaker failed in some way to meet the implicit expectations of the conversation. Each of these two types of breaks will be discussed below.

3.1 Initiating new sequences

Sequences are not always completed satisfactorily. This may in two places within the **o* X2? o* X1 o*** sequence: Before any core move has taken place (when only **o** moves have taken place), and after an **X2** move has occurred but before the responding **X1** move.

student1	o	Okay, so, you know
student2	A2	Careful with that.
student3	A1	Yeah I'm just seeing how it works.
student2	A2	Okay, let's start with one, so
student3	K2	You can do that?
student1	K1	Yeah.
student2	A2	Let's start, then, we need to complete the project.
student1	A1	Yeah yeah yeah.

In the first of these two examples, student1 has attempted to grab the floor to make a statement. The other two students in the discussion ignore this attempt and speak to each other in a clearly unrelated sequence. The **o** move clearly has no relation to the conversation between the other two students, so it should not be included in that sequence.

The second case represents an *inserted* sequence. A suggested action (**A2**) put forward by student2 is questioned by student3; this questioning is responded to by student1. Student2 then restates the previous **A2** action, slightly reformed. Student1 fulfills the sequence by authorizing this suggestion (**A1**). Thus a **K2-K1** sequence occurred in the middle of the **A2-A1** sequence. This does not break Martin's sequence description, it merely forces the conversations to take into account these multiple sequences.

It would be tempting to just leave these threads open to be picked up at any later point in the future. Indeed, there are some cases when a question is only responded to much later in a discourse, after numerous sequences have been initiated and closed. However, this is unwieldy in practice, and there must be some limitation to the number of open threads.

We address this problem by allowing new sequences to be initiated following the decision tree in Figure 3. There must be some limiting point where new threads must cut off old one; we place this limit at six, such that at most six sequences can be active at any one time. When a seventh would be opened through the decision tree process, we close the oldest thread (of the six active threads, this is the thread for which the most recent post is older than the most recent post for the other five threads).

Thus, within a single sequence, the $\mathbf{o}^* X2? \mathbf{o}^* X1 \mathbf{o}^*$ pattern is followed exactly, ignoring challenge moves (described below). Insertion sequences are allowed to switch the topic and then return to the previous topic by maintaining the previous sequence as a separate thread and allowing statements to shift back into those threads. Simultaneous discussions on multiple topics are similarly allowed by simply shifting back and forth between two threads, so long as Martin’s pattern is enforced.

3.2 Challenges

A **ch** move breaks a Martin sequence. The $\mathbf{o}^* X2? \mathbf{o}^* X1 \mathbf{o}^*$ pattern is broken due to a speaker rejecting the authority of the previous speaker to have made the move they made. This can happen for a variety of reasons, but there is a general pattern to challenges, based on the core move that is being responded to.

A **K1** move is challenged by explicitly and directly disagreeing with the factuality of the content of that move. If the **K1** move is a response to a previous **K2** move, the challenge may be formulated as a restatement of the original question, which would reject the authority of the **K1** speaker to answer the question. If the **K1** move is standalone, there must be an explicit textual marker showing that the move is being called inaccurate, such as an explicit “No, you’re wrong” or an accusation of lying.

A **K2** move is challenged by responding with an explicit acknowledgement that you do not have the answer to the question being posed, such as “I don’t know.” This rejects the authority of the questioner to implicitly assume that you have the knowledge being requested.

An **A1** move is challenged by halting or otherwise prompting the narrated action from taking place, for instance, an interjection asking for the action to halt. This takes away the authority of the speaker to make that action.

An **A2** move is challenged by explicit refusal to perform the action being requested. If an **A2** move is merely ignored and a different discussion is continued, this is not a challenge! One reason why this is the case is that it may be an implicit **A1** move - for instance, “Let’s get started” would be an **A2** move, and starting to discuss the topic at hand with a **K** sequence, would not be a challenge of the **A2** just by virtue of not narrating the **A1** move verbally. There needs to be verbal refusal to perform the action in order for a **ch** move to be marked.

4 Example Fragment

Figure 1 gives a 30-line excerpt from a dialogue which exemplifies many of the subtleties of this coding scheme. Graphically, each thread initiation is denoted with a green background in the coded cell; a contribution in blue shows that the code belongs to that thread. A vertical bar in blue shows that while a thread is not active, it is still open to be re-entered. A red asterisk indicates that a thread has been closed to make room for a new thread. Features to note from this excerpt include:

- Note that two threads occur simultaneously, switching back and forth, in the first ten lines. Student2 repeatedly tries to initiate an **A** sequence while Student3 and Student1 go back and forth in a **K** sequence, ignoring Student2.
- When Student3 does respond to Student2, it is a direct and explicit refusal to perform the action being requested. This thus qualifies it to be coded as **ch**.
- The third sequence is a narration of an action by Student3. Because no permission was sought (the action was simply taking place), this is an **A1** move from the start.

1	m4_s2	o			Okay.
2	m4_s2	o			So let's
3	m4_s2	A2			do you want to start doing it?
4	m4_s3		o		You know
5	m4_s1		o		yeah
6	m4_s3		o		Uh, well
7	m4_s2	A2			Okay, can we start the design now?
8	m4_s3		K2		why not three because it will be more secure.
9	m4_s1		o		or maybe
10	m4_s2	A2			Can we start designing?
11	m4_s3	ch			Wait, wait.
12	m4_s2	o			Okay.
13	m4_s3			A1	Let's add, let's up, add up first the final budget.
14	m4_s3			A1	It's ten plus four is fourteen
15	m4_s2		*	o	Yeah.
16	m4_s3	*	K2		and then we need uh how many grocery bags?
17	m4_s2	K2			Uh, one or two, one?
18	m4_s3	K1			uh, no.
19	m4_s2	ch			So one to cut, one to cut and one to put everything into.
20	m4_s3	ch			No.
21	m4_s1	o		*	yeah but oh,
22	m4_s1			K2	what do we use as our
23	m4_s1			K1	ah, so maybe we can start with one.
24	m4_s1		*	K1	And if we need more paper, we can yeah buy more.
25	m4_s2	*	A2		Okay, let's start with one, so.
26	m4_s3	K2			You, can you do that?
27	m4_s1	K1			yeah yeah yeah.
28	m4_s3	ch		*	You don't have to purchase everything at the
29	m4_s1			K2	We don't have to, we don't have to buy everything at once, right experimenter?
30	m4_s3			o	Ah, okay.

Figure 1: A brief excerpt from a three-person dialogue.

- Student3's question in the next sequence brings us to four open conversations, so we must close off the oldest.
- Student2 does not respond to Student3's question with an authoritative answer, but instead opens a new **K2** thread. This insertion sequence counts as a new sequence. While it is on the same topic as the current sequence, it does not fit Martin's pattern.
- Student3's response to Student2 is an explicit rejection, however, it is not a challenge move. The question that Student2 posed in line 17 is phrased as a yes-no question, so it is looking for that type of response. Student3 is not revoking the authority of Student2.
- Student2 does show some authority, however, by rejecting the answer that is given by Student3 in line 18. This rejection is shown as a restatement of the original idea. Because this refusal is a direct flouting of the answer given in the previous line, it is coded as **ch**.
- Student3 now responds again with another "No." However, this time the response is not to a yes-no question but instead to an assertion phrased as a primary authority. This makes the outright rejection a more clear **ch** move.
- Student3 starts another insertion sequence in line 26, and just as Student2 did previously, repeats the question after getting an answer, which implicitly rejects the authority of the other students to give the answer they gave. Thus this is another **ch** move.

5 Summary

The coding scheme that we have presented above therefore follows these maxims:

- Utterances can be coded as one of four core moves, **K1**, **K2**, **A1**, and **A2**, as well as **ch** for direct challenges and **o** for all other moves.
- Whenever possible, sequences should fit the **o* X2? o* X1 o*** pattern from Martin.
- When this pattern is broken, the previous thread remains open and a new thread is created, closing off an old thread if necessary.
- Threads can be intertwined and can be returned to later if they are explicitly being recalled or responded to.
- Many threads might be introduced in rapid fire succession if ideas fail to take off or a group fails to act cohesively and cooperatively. This is not a problem with the system.
- The two decision trees in the next two pages give explicit instructions for any time when the correct action to take is not clear.

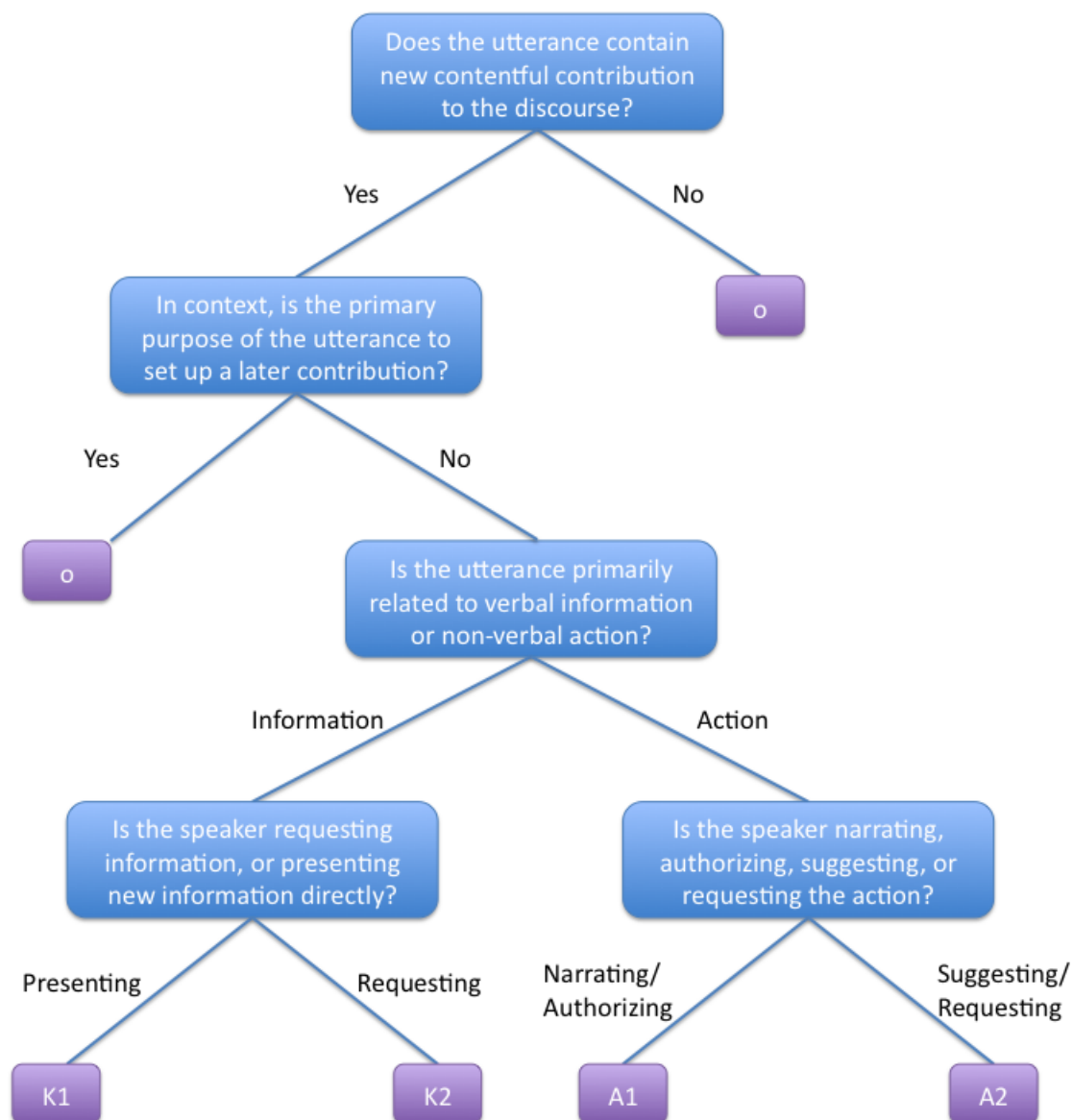


Figure 2: Decision tree for core move labelling.

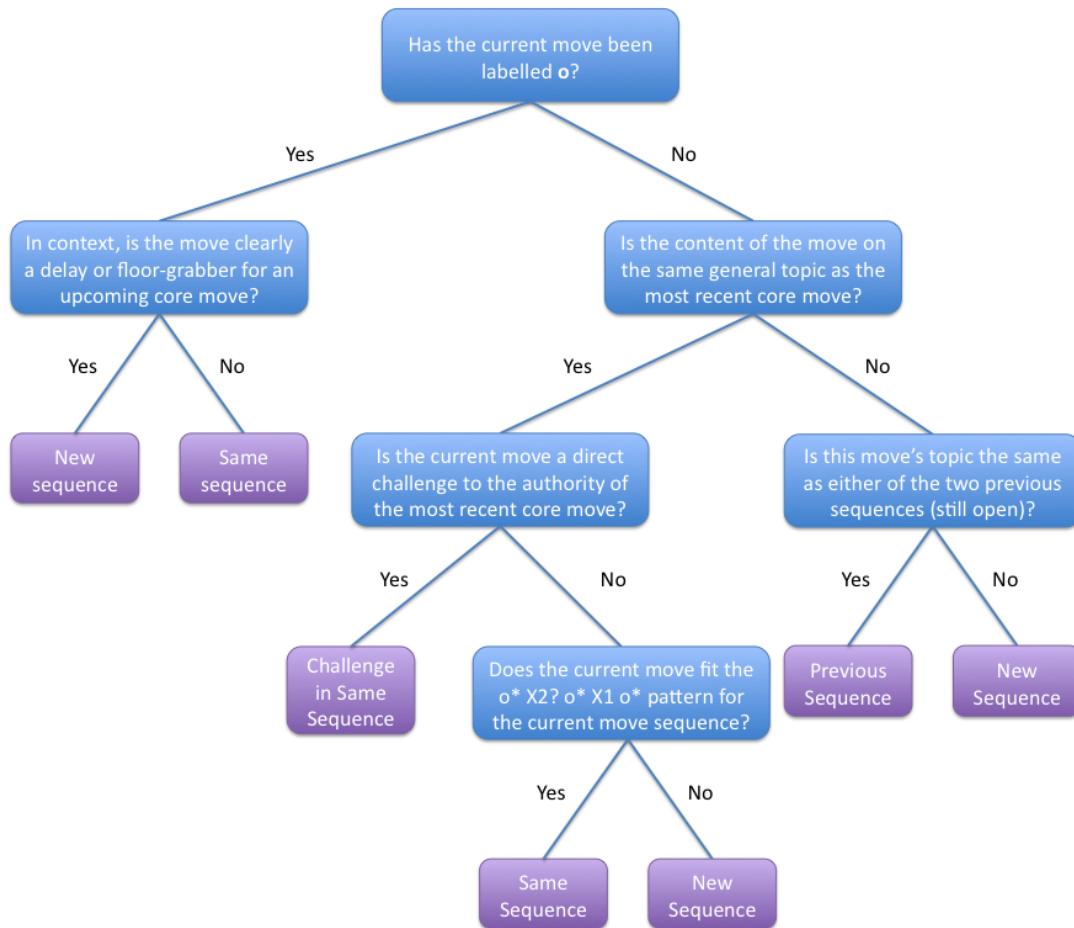


Figure 3: Decision tree for new sequence initiations.