

Ibn Sīnā on analysis: 1. Proof search. Or: Abstract State Machines as a tool for history of logic

Wilfrid Hodges
Herons Brook, Sticklepath,
Okehampton, Devon EX20 2PY, England
wilfrid.hodges@btinternet.com

DRAFT 6 September 09. This is a paper submitted for a Festschrift,
and I have removed some personal references.

Abstract

The 11th century Arabic-Persian logician Ibn Sīnā (Avicenna) in section 9.6 of his book *Qiyās* gives what appears to be a proof search algorithm for syllogisms. We confirm that it is indeed a proof search algorithm, by extracting all the essential ingredients of an Abstract State Machine from Ibn Sīnā's text. The paper also contains a translation of the passage from Ibn Sīnā's Arabic, and some notes on the text and translation.

1 Introduction

This paper contains a translation and commentary on section 9.6 of Ibn Sīnā's major work on logic, the volume 'Syllogism' (*Qiyās*) from his encyclopedic *Šifā'*, a work written in Arabic in the 1020s. The section is itself a loose commentary on some lines in Aristotle's *Prior Analytics* i.32. It falls into two parts. In the first part Ibn Sīnā describes what he sees as the task of logical 'analysis' (*taḥlīl*). One ingredient of that task is to complete formal proofs which have a piece missing, and Ibn Sīnā gives his account of this in the second part. A special case of this problem (though not one mentioned by Ibn Sīnā himself) is to find a formal proof where everything is missing except the conclusion, and this is precisely the task of proof search.

As far as I know, Ibn Sīnā's account is the first work to come anywhere near describing a proof search algorithm in formal logic.

Abstract State Machines (ASMs [6]) were introduced by Yuri Gurevich [10]. They give a framework for describing algorithms with complete precision at whatever level of refinement we choose. The main business of this paper is to describe an algorithm. The fact that Ibn Sīnā himself is less than explicit about some details is no excuse for us to lapse into vagueness. If we want to record with decent precision what Ibn Sīnā used or understood, and what he didn't, we need the best descriptive tools; and so I turned to ASMs. Fortunately the work is already partly done, because a famous early application of ASMs was Börger and Rosenzweig's specification of the proof search algorithm of Prolog to meet the ISO 1995 standard [5].

As far as I know, the use of ASMs below is the first application of ASMs to the history of logic, and one of the first applications of ASMs in the humanities. (A recent paper [11] calls for applications in linguistics, but these go in a rather different direction.) In practice the task of constructing an ASM was an invaluable research tool; it kept raising questions to be addressed to Ibn Sīnā's text. Remarkably often Ibn Sīnā does answer the questions in his text, though I often had to refer to other sections of the *Qiyās* for clarifications. I doubt that any other logician between Aristotle and Leibniz would have come through this test as successfully as Ibn Sīnā does.

The paper has an unusually wide spread of prerequisites. First there is the Arabic text of Ibn Sīnā and its historical background. Second there are the mathematical facts about syllogisms. Third there is the methodology of Abstract State Machines. Unfortunately papers are linear strings of text, so some prerequisites will have to wait their turn. The structure of the paper is as follows:

Section 1. Introduction.

Section 2. Historical background (on logic from Aristotle to Ibn Sīnā).

Section 3. *taḥṣīl* (roughly the counterpart in Ibn Sīnā of Tarski's notion of setting up a deductive theory).

Section 4. Mathematical prerequisites on syllogisms.

Section 5. Extracting the algorithm.

Section 6. Review.

Appendix A. Translation of *Qiyās* 9.6.

Appendix B. Notes on the text translated.

Appendix C. The ASM.

The passage translated in Appendix A (*Qiyās* 9.6) needs to be matched up with *Qiyās* sections 1.5 on categorical syllogisms, 9.3 on compound syllogisms, 9.4 on supplying missing premises of simple syllogisms and 9.7–9 on other aspects of analysis. I will do my best to get translations of these sections onto my website at <http://wilfridhodges.co.uk>. Meanwhile Tony Street [30] gives a useful summary of Ibn Sīnā's theory of predicative syllogisms.

I thank Egon Börger, Jamal Ouhalla, Roshdi Rashed and Gabriel Sabagh for some valuable remarks, and Amirouche Moktefi for advice on the Arabic translation. But I take full responsibility for errors; there are bound to be some, though I believe the use of ASMs has eliminated many of the more serious ones.

2 Historical background

In the mid 4th century BC, Aristotle noticed that many arguments in mathematics, metaphysics and elsewhere have one of a small number of forms, and that any argument of any of these forms is guaranteed to be convincing. He referred to arguments of these forms as 'syllogisms', and he classified them into three 'figures'. He listed and discussed the argument forms in lectures or writings which have reached us as a book called *Prior Analytics* [3]. That book was part of the edition of Aristotle's writings which was put together by Andronicus in the first century BC. Apart from the text itself, we have virtually no evidence of what Andronicus did with his raw materials. He may have put things together in ways that Aristotle never intended.

Andronicus' edition of Aristotle came to form a collection of textbooks for the offspring of cultured parents in the Roman Empire. By the late second century AD it had become clear that some explanatory commentaries on Aristotle's text were needed, and Alexander of Aphrodisias wrote a set. His commentaries were followed by many others, mostly now lost. The two surviving Roman Empire commentaries on the parts of the *Prior Analytics* that will concern us are those of Alexander and the 6th century Alexandrian scholar John Philoponus. (See Ebbesen [8] Chapter III for an account of the intellectual climate in which commentaries on Aristotle's logic arose.)

By the middle of the 8th century the new Arab empire had started to absorb western scholarship, including Aristotle's logic. Ibn Sīnā reports

that in the 990s he visited the large library of the Sultan of Bukhara (in present-day Uzbekistan), and found that it contained a catalogued collection of books of ‘the ancients’, which included a number of rare items, presumably in Arabic or Persian translation (Gutas [12] p. 28f). Most of this material has now gone missing, together with the Greek originals. In the 10th century Al-Fārābī wrote a lengthy Arabic commentary on the *Prior Analytics*. Ibn Sīnā probably knew this work, but today very little of it survives, and nothing that will help us in this paper.

The longest and fullest of Ibn Sīnā’s writings in logic is the Logic section of his encyclopedic *Šifā’*, written in the 1020s. It takes the form of a commentary on Aristotle’s logic, some of it very close to Aristotle and some of it apparently quite new. In *Šifā’* the book *Qiyās* (‘Syllogism’ [15]) is his commentary on the *Prior Analytics*. In *Qiyās* the section 9.6 is his commentary on just twenty-three lines of the *Prior Analytics*, namely 46b40–47a22. These are the opening lines of section i.32 of *Prior Analytics*.

Aristotle begins these twenty-three lines by announcing that his next task is to ‘explain how we can lead deductions back into the figures stated previously’ ([3] p. 50). He adds that this is a matter of analysing (*analúoimen*) arguments that ‘have already been produced’ ([3] p. 50) into the three syllogistic figures. The arguments could have been already produced ‘in writing or in speech’ ([3] p. 51). He makes it clear that analysis includes both identifying the underlying form of an argument, and also *repairing* the argument, for example adding missing premises or removing redundancies.

Alexander of Aphrodisias and Philoponus both report Aristotle’s views faithfully. Alexander adds that since the whole book is called *Analytics*, this section on ‘analysis’ of arguments must be the heart of it. The modern commentator David Ross agrees that the use of ‘analyse’ in this passage is the source of the name *Analytics*, both for this book and for Aristotle’s work *Posterior Analytics* on the theory of knowledge ([28] p. 400). He also calls attention to the mathematical use of *analúein* to mean working backwards from conclusions to premises. This usage agrees with the part of Aristotle’s ‘analysis’ that consists of finding a missing premise.

Ibn Sīnā’s section 9.6 falls into two parts. The first part, from paragraph [9.6.1] to [9.6.5], more or less matches Aristotle’s text. The second part, consisting of paragraphs [9.6.6] to [9.6.12], is completely new. It picks up Aristotle’s brief remark that the syllogism being analysed may have a premise missing, and it proposes a systematic way of repairing the hole. It also expresses the outcome in terms of making the syllogism ‘*ḥāṣil*’. This is a central notion in Ibn Sīnā’s methodology, as follows.

3 *taḥṣīl*

There are two notions to be brought together here. One is *taḥlīl*, which is the Arabic word that Ibn Sīnā uses to translate Aristotle's *análusis*. Ibn Sīnā regarded *Prior Analytics* i.32–46 as a manual of analysis, and he commented on these sections in sections 9.6 to 9.9 of his *Qiyās*. The material in sections 9.7–9 is not directly related to that in 9.6, but it is needed for a full picture of Ibn Sīnā's understanding of analysis.

The second notion is *taḥṣīl*, which means 'making *ḥāṣil*'. There are no easy English translations of *taḥṣīl* and *ḥāṣil*, and even if there were, we would still need to explain how the notions fit into Ibn Sīnā's view of philosophical activity.

At the most literal level, *ḥāṣil* means 'available for use', so that *taḥṣīl* means 'making available for use'. The word *ḥāṣil* occurs nine times in *Qiyās* 9.6, and its grammatical relatives many times more. A thing is *muḥaṣṣal* if it has been made available for use. Here is a remarkable example of the literal usage:

- ... some people demonstrate without any rule, like Archimedes
(1) who demonstrated mathematically, since in his time logic wasn't yet available (*lam yakun muḥaṣṣal*). (*Qiyās* [15] 15.10f.)

Ibn Sīnā has his history confused — Archimedes was born a hundred years later than Aristotle. The idea that Archimedes demonstrated 'without any rule' is puzzling. Roshdi Rashed (personal communication) suggests that the point is that geometrical reasoning, of the kind that Archimedes used, is not algorithmic.

For Ibn Sīnā, one of the main tasks of a philosopher was to apply *taḥṣīl* to the ideas of earlier philosophers. He refers several times to commentators on Aristotle as *muḥaṣṣilūn*, people who make *ḥāṣil*. A typical example is in *Iṣārāt*:

- Nothing but this has been stated by earlier scholars (*muḥaṣṣilūn*),
(2) but in a manner overlooked by recent ones. ([20] I.9.2, p. 150 of Inati.)

Likewise at *Najāt* [18] i p. 35.4 he refers to 'Alexander and a number of later *muḥaṣṣilūn*'. What were the commentators doing that counted as 'making *ḥāṣil*'? The answer has to depend on exactly what they were making *ḥāṣil*. We find places where Ibn Sīnā describes the following things as being made *ḥāṣil*: (a) concepts, (b) propositions, (c) syllogisms, (d) knowledge. Usages (b) and (c) are frequent in *Qiyās* 9.6.

Usage (d) is illustrated by the following passage near the beginning of *Burhān*:

- (3) Knowledge — whether it is obtained through reflective reasoning (*fikr*) or is *ḥāṣil* without being obtained through reflective reasoning — is of two kinds. One of them is assent (*taṣḍīq*) and the other is conceptualisation (*taṣawwur*). Knowledge in the form of assent, when it is obtained through reflective reasoning, becomes *ḥāṣil* to us through a syllogism. Knowledge in the form of conceptualisation, when it is obtained through reflective reasoning, becomes *ḥāṣil* to us through a definition. ([16] 3.10–12.)

Here Ibn Sīnā sets out two independent classifications of kinds of knowledge. The first classification is into those forms of knowledge which depend on reflective thinking and those which come to us without our having to think reflectively. The second classification, which is fundamental throughout Ibn Sīnā's logic and epistemology, is between two processes that lead to knowledge. The first of these processes is conceptualisation (*taṣawwur*); it leads us to having a concept, and Ibn Sīnā counts this as a kind of knowledge. The second process is assent (*taṣḍīq*), i.e. coming to recognise that a proposition is true; it leads to knowledge of the fact stated by the proposition. Although Ibn Sīnā in his first sentence uses *ḥāṣil* only for knowledge not dependent on reflective thinking, the rest of his text shows that this is just an accident of style, and both kinds of knowledge can be *ḥāṣil*. In fact the passage suggests that for knowledge, being *ḥāṣil* and being 'obtained' amount to the same thing.

The passage gives us strong clues about usages (a) and (b), because *taṣawwur* leads to knowledge of concepts and *taṣḍīq* leads to knowledge of propositions.

Take concepts first. Here Ibn Sīnā's usage slots in with a philosophical usage that had been around already for many decades. The 9th century translator of Aristotle, Ishāq bin Ḥunain, rendered Aristotle's 'indefinite' (*áoristos*) as *gair muḥaṣṣal*, i.e. 'not *muḥaṣṣal*' (*Peri Hermeneías* 16b14, translated at [21] p. 111). The implication is that a concept is *muḥaṣṣal* if it is determinate or well-defined. Kutsch [23] assembles a large number of references where *muḥaṣṣal* means this.

In (3) above, Ibn Sīnā is saying that a concept is made determinate or well-defined by being given a definition. He certainly regarded this as one of the main tasks of the Aristotelian commentators. At *ʿIbāra* [14] 2.9f he refers to 'those commentators who are experts on definition' (*al-muḥaṣṣilūn min 'ahl ṣinā'a t-taḥdīd*). Incidentally there is a close analogy here with Ernst

Zermelo's notion of a 'definit' criterion for class membership in mathematics [33]. Just as Ibn Sīnā expected a commentator to define in genus-differentia form, so Skolem proposed that Zermelo's 'definit' should be read in set theory as 'first-order definable'.

We turn to propositions. Ibn Sīnā says in several places that we can't assent to a proposition until we have conceptualised its meaning, i.e. until we understand it. Thus in *Easterners*:

- So when the conceptualisation is made *ḥāṣil* for us, assent to [the proposition] is made *ḥāṣil* for us [too]. But the conceptualisation
- (4) comes first; so if we don't conceptualise a meaning then we don't get assent to [the proposition]. Sometimes we get the conceptualisation without assent attached to it. ([19] 9.12–14.)

So there is a sense in which making a proposition *ḥāṣil* is like making a concept *ḥāṣil*; we clarify the construction of the proposition and the meanings of the words in it. But in both (3) and 4, Ibn Sīnā mentions another sense in which a proposition can become *ḥāṣil*, namely that we come to recognise that it is true. In (3) he says that this happens when we deduce the proposition through a syllogism. We note that for this to work, the premises of the syllogism must already be *ḥāṣil* in this sense.

Ibn Sīnā regarded this second kind of making propositions *ḥāṣil* as central to the activity of philosophical commentators. The activity consists of taking a claim made by Aristotle (for example), or on his behalf, and looking to see how much of an argument is offered to support the claim. Then one works on the argument to fill in gaps, remove irrelevances etc. etc. In fact one performs exactly the 'analysis' that we saw Aristotle himself describing in *Prior Analytics* i.32ff. But while for Aristotle and Alexander this kind of analysis was one of the general tools of logic, Ibn Sīnā thought he could point to a large body of published work specifically devoted to it, namely the philosophical commentaries. (There is a hint of this view already in Philoponus [24] p. 315 l. 20, where he says that the syllogism to be analysed may come from 'the ancients'.)

To fill in the history a little, the idea of commenting on a philosopher by reducing that philosopher's arguments to syllogistic form seems to have surfaced first among the Middle Platonist commentators on Plato's dialogues in the first century AD. It may have been encouraged by a desire to show that Plato was just as good a logician as Aristotle (a view that Ibn Sīnā explicitly rejects with contempt [17] pp. 114f). For example Alcinous [1] 158.42–159.3 finds the following second-figure syllogism in Plato's *Par-*

menides 137d8–138a1:

- A thing that has no parts is neither straight nor circular. A thing
(5) that has a shape is either straight or circular. Therefore a thing
that has no parts has no shape.

(The second premise is obviously false. In any case Plato as I read him gives ‘straight’ and ‘circular’ as typical examples of shapes, not as an exhaustive list. But Alcinous wasn’t the world’s greatest logician.)

Most of the surviving Roman Empire or Arabic commentaries on Aristotle, including those of Ibn Sīnā, do contain explicit reductions of particular arguments to syllogistic form. These reductions form a very small proportion of the text of the commentaries. But probably Ibn Sīnā regarded it as a criterion of the quality of a commentary that it should be straightforward to analyse the commentator’s arguments into syllogistic form. The analogy with modern set theory applies here too. We don’t expect set theorists to set out their arguments as first-order deductions from the Zermelo-Fraenkel axioms, but we do take it as a criterion of a sound set-theoretic argument that it should be routine to reduce the argument to this form.

By implication we have already said what it should mean to describe a syllogism as *ḥāṣil*. We make a syllogism *ḥāṣil* by analysing it into a form so that it makes its conclusion *ḥāṣil*. This involves putting it into one of the standard syllogistic moods, and ensuring that its premises are *ḥāṣil*.

There are a couple of nuts-and-bolts points about *ḥāṣil* that can be made here as well as anywhere. First, the notion of *ḥāṣil* is relational: a thing can be *ḥāṣil* for me but not for you. This is explicit in both (3) and (4). As far as I’m aware, there is no notion in Ibn Sīnā of a thing being ‘*ḥāṣil* in itself but not for us’, such as we might expect in 13th or 14th century Scholastics.

And second, the set of propositions that are *ḥāṣil* for you is dynamic: you can add new items to the set by deducing them from things already in the set. This causes some problems of terminology. In proof search we assume we have a database T of sentences, and we search for proofs of given sentences from assumptions that are in T . In Ibn Sīnā’s case the set T is the set of propositions that are *already ḥāṣil*. But it’s natural for him to say that a successful proof search makes another proposition ϕ *ḥāṣil*, and it could look as if he is saying that ϕ is added to the database. Granted, Prolog has a function *assert* which does exactly that. But adding ϕ to T is completely different from deducing ϕ from things already in T , and it’s the latter that is important for the proof search algorithm. The remedy is to distinguish strictly between those propositions that were *already ḥāṣil* and those that *become ḥāṣil* through application of the algorithm. Ibn Sīnā’s choice of words

doesn't always help us to make this distinction; see Problem 32 and the note on it.

4 Mathematical prerequisites on syllogisms

4.1 Syllogistic sentences

In *Qiyās* [15] section 9.6 Ibn Sīnā discusses four kinds of sentence. He gives Arabic examples, which I translate in (6) below. But for the logical theory the Arabic renderings are less important than the truth conditions which he sets out in *ʿIbāra* [14] section 2.2. He sometimes uses other Arabic wordings for the same sentence types (see for example the note on Problem 14).

The four sentence types are as follows, together with their names and their truth conditions:

- Universally quantified affirmative, 'Every A is a B '. This counts as true if there are A s, but there are no A s that are not B s, and false otherwise.
 - Universally quantified negative, 'No A is a B '. This counts as true if there are no A s that are B s, and false otherwise.
- (6)
- Existentially quantified affirmative, 'Some A is a B '. This counts as true if there is some A that is a B , and false otherwise.
 - Existentially quantified negative, 'Some A is not a B '. This counts as true in two cases: (a) there is an A that is not a B , and (b) there are no A s. Otherwise it counts as false.

The letters ' A ' and ' B ' are place-holders for two distinct 'terms' (*ḥadd*). For present purposes we can think of terms as being the meanings of actual or possible common nouns. Ibn Sīnā believed that when reasoning we manipulate terms in our minds through linguistic expressions that mean them. This allowed him to do the same in his logical theory, for example using common nouns as surrogates for their meanings. A syllogistic sentence of the form 'Every A is a B ' is got by putting common nouns in place of ' A ' and ' B ', with the sole restriction that the two common nouns must have different meanings. By 'syllogistic sentence' we will mean a sentence of one of the four forms in (6).

A syllogistic sentence can be identified by four features. The first is the ‘subject’ (*mawḍūʿ*), which is the term put for ‘*A*’. The second is the ‘predicate’ (*maḥmūl*), which is the term put for ‘*B*’. The third is the ‘quantity’ (*kam*), which is either ‘existentially quantified’ (*juzʿī*) or ‘universally quantified’ (*kullī*). The fourth is the ‘quality’ (*kaifa*), which is either ‘affirmative’ (*mūjib*) or ‘negative’ (*maslūb*). For purposes of the ASM I treat a syllogistic sentence as a 4-tuple

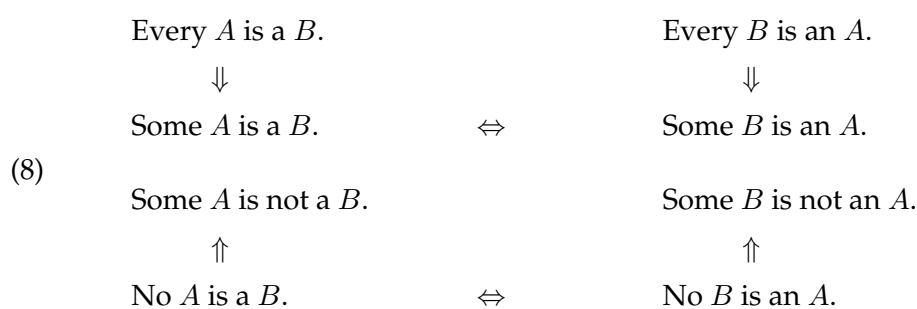
(7) [subject,predicate,quantity,quality]

using 0 for existentially quantified and affirmative, and 1 for universally quantified and negative. (See (Def1) in Appendix C.)

The conditions for ‘Every *A* is a *B*’ to be true are satisfied exactly when those for ‘Some *A* is not a *B*’ are not satisfied. So each of these syllogistic sentences means the same as the negation of the other. We say they are ‘contradictories’ of each other, and we write $\bar{\phi}$ for the contradictory of ϕ . Likewise ‘No *A* is a *B*’ and ‘Some *A* is a *B*’ are contradictories.

By ‘formal sentences’ I mean the expressions that we get if we put uninterpreted 1-ary relation symbols (we call them ‘term symbols’) in place of ‘*A*’, ‘*B*’ in (6) above. The truth conditions translate at once into conditions for a formal sentence to be true in a structure. So we have a model-theoretic notion of entailment: a set *T* of formal sentences entails a formal sentence ψ if and only if there is no structure in which all the formal sentences in *T* are true but ψ is not true. Though this notion was unknown to Ibn Sīnā, it gives us some mathematics that will be helpful for understanding various things that Ibn Sīnā does.

For example it allows us to demonstrate all the cases where one formal sentence entails another. They are as follows (where we write \Rightarrow for ‘entails’):



The top and bottom halves of this diagram are not independent. Each sentence in the bottom half is the contradictory of its counterpart in the top

half. Hence the arrows in the bottom half go the opposite way to those in the top half. Ibn Sīnā recognised all the instances of these entailments as examples of ‘following from’.

4.2 Inconsistent sets

A set T of formal sentences is ‘consistent’ if there is a structure in which all the formal sentences in T are true, and ‘inconsistent’ if there is no such structure. It is ‘minimal inconsistent’ if it is inconsistent but every proper subset of it is consistent.

We can characterise the minimal inconsistent sets of formal sentences as follows. First, by a ‘minimal circle’ we mean a set of formal sentences arranged in a circle

$$(9) \quad [\phi_1, \dots, \phi_n] \quad (n \geq 2)$$

where ϕ_1 is immediately after ϕ_n in the circle, in such a way that every term symbol appearing in the sentences occurs exactly twice, and the two occurrences are in adjacent sentences in the circle.

Theorem 1 *Every minimal inconsistent set of formal sentences can be arranged into a minimal circle.*

We say that a term symbol t in a formal sentence ϕ is either ‘distributed’ or ‘undistributed’ in ϕ as follows. If t is subject of ϕ then t is distributed in ϕ if ϕ is universally quantified, and undistributed otherwise. If t is predicate of ϕ then t is distributed in ϕ if ϕ is negative, and undistributed otherwise.

Theorem 2 *A minimal circle C is inconsistent if and only if it meets the following two conditions:*

1. *Each term occurring in sentences of C has at least one distributed occurrence.*
2. *Exactly one of the sentences in C is negative.*

These two theorems are equivalent to results in §46 of Thom [31], which Thom proves proof-theoretically. But they can be proved directly from the truth conditions in (6). Ibn Sīnā himself probably knew Theorem 1 from experience, though it’s hard to see how he could have proved it. On the other hand he almost certainly didn’t know Theorem 2. Any form of this result involves partitioning occurrences of terms in syllogistic sentences into the

two classes that we called distributed and undistributed, and no such partition has been found in Ibn Sīnā's logical writings.

Now given an inconsistent circle as in (9), we can take out any one sentence, say ϕ_i . Then the remaining sentences entail $\overline{\phi_i}$; moreover all entailments between formal sentences, where there are no redundant sentences in the entailing set, are formed in this way. List the entailing sentences in their order in the circle:

$$(10) \quad [\phi_{i+1}, \dots, \phi_n, \phi_1, \dots, \phi_{i-1}]$$

Then the sequence (10) has the property that every term symbol occurs twice, in two adjacent sentences of the sequence, except for one term symbol that occurs only in the first sentence and another one that occurs only in the last sentence. We describe a sequence (10) with this property as a 'linkage' (*qarīna*, though strictly Ibn Sīnā uses the term only for such sequences of length 2). The sequence (10) and the sentence $\overline{\phi_i}$ together form a 'formal separated syllogism' whose 'premises' (*muqaddamāt*) are the sentences in (10) and whose 'conclusion' (*natīja*) is the sentence $\overline{\phi_i}$. The expression 'separated syllogism' (*qiyās mafṣūl*) is from Ibn Sīnā (*Qiyās* [15] p. 436.1), though strictly he uses it only when there are more than two premises.

So we can speak of a 'separated syllogism', meaning an entailment between syllogistic sentences, got by taking a formal separated syllogism and replacing the distinct term symbols by distinct terms. The separated syllogisms that Ibn Sīnā recognises all have the property that their premises entail their conclusion (model-theoretically); in his terminology the conclusion 'follows from' (*yalzam*) the premises. But later in this section it will take us some time to unpick the relationship between Ibn Sīnā's notion of following from and our notion of entailment.

But first we turn to the notion that the proof search algorithm is meant to deal with: separated syllogisms with a premise missing. Suppose for example that we have a separated syllogism with premises $[\phi_1, \dots, \phi_m]$ and conclusion χ , and we remove one or more adjacent premises, say ϕ_j and ϕ_{j+1} . In the inconsistent circle the contradictory of χ belongs at the beginning or the end; we will put it at the end:

$$(11) \quad [\phi_1, \dots, \phi_{j-1}, \phi_{j+2}, \dots, \phi_m, \bar{\chi}].$$

Now we can describe the gap as follows. It comes immediately after the $(j - 1)$ -th sentence in the sequence (11); we call the number $j - 1$ the 'gap site'. If ϕ_1 and ϕ_2 had been removed, the gap would be immediately after

$\bar{\chi}$, which is the $(m - 1)$ -th sentence in $[\phi_3, \dots, \phi_m, \bar{\chi}]$, so the gap site would be $m - 1$. Also when the linkage (11) contains at least two sentences, there is a unique term shared by the lefthand missing sentence and the one to the left of it; we call this term the ‘left edge’ of the gap. Likewise there is a unique term shared by the righthand missing sentence and the one to the right of it in (11); we call this the ‘right edge’ of the gap.

Thus in Problem 20 Ibn Sīnā gives the following example:

- (12) Conclusion (understood from Problem 12) ‘Some C is not an A ’.
 Premises ‘Some D is a C ’ and ‘No A is a B ’.

Putting the contradictory of the conclusion at the end gives the sequence

- (13) [Some D is a C ’, ‘No A is a B ’, ‘Every C is an A ’].

The gap site is 1, the left edge is D and the right edge is B . The definitions just given are more formal than Ibn Sīnā himself uses. But he provides several types of example with different gap sites. At Problem 7 he uses the left and right edges of the gap. In this problem he does also include an irrelevant term, and clearly he knows that it’s irrelevant; perhaps he wants to encourage the student to work out that only the left and right gaps are needed at that stage in the algorithm. (See the notes on Problem 7.)

Ibn Sīnā doesn’t consider the case where all the premises are missing — which is actually the case that corresponds to the proof search problem for Prolog. In this case the gap comes immediately after the contradictory of the conclusion, so the gap site is 1. But with only one sentence present, there is no way of telling which of its terms is the left edge and which is the right. We need a definite choice; I stipulate that in this case the left edge is the subject of the conclusion and the right edge is its predicate. (This is not quite arbitrary; it reconciles two messages that Ibn Sīnā sends about which end of the gap to start with when we fill it. Namely in *Qiyās* [15] section 9.3 he works from the left side to the right, and when finding middles in section 9.4 he starts with the subject of the conclusion.)

Identifying the gap site and the left and right edges is necessary for the algorithm, so I made it a module of the ASM. See (ASM3) in Appendix C for the module DESCRIBE. I haven’t bothered to spell out the formal definition in cases like this where there is a purely book-keeping manipulation that can be specified unambiguously in English.

4.3 Simple syllogisms

A ‘simple syllogism’ (*qiyās basīṭ*) is a separated syllogism with two premises. Ibn Sīnā often abbreviates this to ‘syllogism’.

The term which occurs in both premises of a simple syllogism is called the ‘middle’ (*wast*). The term which is subject of the conclusion is called ‘lesser’ (*asḡar*), and the premise containing it is the ‘minor’ (*ṣuḡrā*). The term which is predicate of the conclusion is called ‘greater’ (*akbar*), and the premise containing it is the ‘major’ (*kubrā*).

Buried in the definition of ‘simple syllogism’ there is the condition that the conclusion follows from the premises. But we remarked earlier that Ibn Sīnā’s notion of ‘follows from’ is some way distant from the model-theoretic definition of entailment. In the case of two-premise arguments he doesn’t recognise as syllogisms any that are not model-theoretically valid. But he puts two further requirements, as follows.

Sometimes Ibn Sīnā speaks as if a syllogism consists of just the premises. For this terminology to work, there has to be a unique way of reading off the conclusion from the premises. By Theorem 3 below, if two syllogistic premises do entail a syllogistic conclusion, then there is a strongest conclusion that they entail. But there are two cases where this strongest conclusion is not uniquely determined, namely ‘Some *A* is a *B*’ (which is logically equivalent to ‘Some *B* is an *A*’) and ‘No *A* is a *B*’ (which is logically equivalent to ‘No *B* is an *A*’). In these two cases Ibn Sīnā resolves the question by the following condition:

Premise order condition The minor premise is listed before the major premise.

In other words, the subject of the conclusion is the term that occurs in the first premise. In fact Ibn Sīnā follows this rule uniformly for all simple syllogisms, even where there is no ambiguity to be resolved.

Thus for example at *Qiyās* [15] 114.6 he gives the mood *Cesare* in the form

(14) Every *C* is a *B* and no *A* is a *B*, so no *C* is an *A*.

while at 115.17 he cites *Camestres*:

(15) No *C* is a *B* and every *A* is a *B*, so no *C* is an *A*.

One of the very few counterexamples to this convention is in Problem 3 below, where he infers ‘No *C* is a *D*’ from ‘Every *D* is a *B*’ and ‘No *C* is a *D*’ in that order. This is probably an accident of his exposition; see the note on that Problem.

Ibn Sīnā also imposes a further condition, which rules out what are sometimes known as ‘fourth figure syllogisms’:

Fourth figure condition The middle term is not both predicate of the first premise and subject of the second premise.

At *Qiyās* 107.12 he describes arguments that violate the fourth figure condition as ‘unnatural, unacceptable and unsuitable for the practice of serious study’.

With the help of Theorem 2 one can show that a model-theoretically valid simple syllogism which fails the fourth figure condition must have a premise of one of the following three forms: ‘Some A is a B ’, ‘No A is a B ’, or ‘Every A is a B ’ where the syllogism would still be model-theoretically valid if we replaced this premise by ‘Some A is B ’. So it’s always possible to bring such a syllogism into a form that Ibn Sīnā accepts, by using the implications in (8). This will involve a ‘conversion’ (*aks*), which swaps the order of the two terms in one premise. Ibn Sīnā notes at Problem 59 that a positive solution for the problem is impossible unless one makes a conversion in the premise. At Problems 29 and 44 he comments that conversion makes no difference to the outcome. Note also his remark about conversion at 466.3f.

Following Aristotle, Ibn Sīnā classifies the possible shapes of simple syllogisms into three figures; the first and second figures have four shapes each, called ‘moods’, and the third figure has six ‘moods’. Ibn Sīnā expects his students to know this catalogue by heart. In fact at 466.4ff he says that a student who hasn’t memorised the catalogue is not going to be able to follow the algorithm. At first sight this is puzzling, because his account of the algorithm doesn’t ever seem to use the figures and moods. Closer inspection reveals one hidden reference to ‘first [figure]’ in Problem 3, and this reference shows what is going on. Ibn Sīnā expects his students to be able to recognise, given two syllogistic sentences ϕ_1 and ϕ_2 , whether there is a syllogism with these as its premises; and where there is such a syllogism, to state its conclusion. In the style of education that he favours, the students memorise this information, and he expects them to do it in terms of figures and moods. (But don’t assume that he would have used Theorem 2 if he had known it. We come back to this in subsection 6.2 below.)

To carry out the proof search algorithm, the student needs to be able to find, for any pair of sentences $[\phi_1, \phi_2]$ that are the premises of a syllogism, the strongest consequence of these premises. In our ASM the function *consequence* will perform this operation; we leave it to the implementer to decide how to compute the values of the function. See (Def3) in Appendix C. A sequence of sentences that are not the premises of any separated syl-

logism is said to be ‘sterile’ (*‘aqīm* — again we generalise Ibn Sīnā’s usage from two premises to any number). When $[\phi_1, \phi_2]$ is sterile, we give *consequence*(ϕ_1, ϕ_2) the formal value *sterile*.

We mentioned a theorem about strongest consequences. It says the following:

Theorem 3 *Let T be a consistent set of formal sentences and C the set of all formal sentences ψ such that T entails ψ and there is no proper subset of T that entails ψ . Then if C is not empty, C contains a sentence ψ which entails all the other sentences in C .*

Theorem 3 can be proved from Theorem 2. The sentence ψ in the conclusion of Theorem 3 is what we have been calling the ‘strongest consequence’ of T ; it’s unique up to the equivalences in (8).

For simple syllogisms, i.e. the case where T has size 2, Theorem 3 seems to have been common knowledge in Ibn Sīnā’s time, and it would have been easy to prove by enumerating the possible cases. Probably the better logicians had a shrewd idea that it was true for any size of T , but I don’t recall seeing it stated in the middle ages, and I doubt they could have proved it.

By cutting the inconsistent circle at a different place, Theorem 3 yields a corollary:

Corollary 4 *Let T be a consistent set of formal sentences and ψ a formal sentence. Let K be the set of all sentences χ such that $T \cup \{\chi\}$ entails ψ but there is no proper subset T' of T such that $T' \cup \{\chi\}$ entails ψ . Then if K is not empty, it contains a sentence χ which is entailed by each of the other sentences in K .*

We call the sentence χ in the conclusion of Corollary 4 the ‘weakest fill’; it’s unique up to the equivalences in (8).

4.4 Connected syllogisms

Ibn Sīnā never attempts to apply any definition of ‘follows from’ directly to separated syllogisms with more than two premises. For simple syllogisms he understands ‘follows from’ in terms of how our minds manipulate ideas, and it would hardly be plausible to assume that we could hold in our minds a set of a thousand premises. Instead he maintains that a separated syllogism is shorthand for a more complex kind of syllogism, namely a tree of simple syllogisms. At *Qiyās* [15] p. 436.1 he describes such

a tree as a ‘connected syllogism’ (*qiyās mawṣūl*). He explains at *Qiyās* [15] p. 442.8 that separated syllogisms are so-called because in them the intermediate conclusions (the conclusions of all the simple syllogisms except the one at the root of the tree) are separated from the premises (presumably he means the premises at the leaves of the tree), so that the premises are mentioned explicitly but the intermediate conclusions are left out. At *Burhān* [16] 141.15ff he comments that a connected syllogism with a thousand intermediate steps is no big deal provided we are ‘mentally prepared for the drudgery’.

So part of the job of analysis is to find these intermediate conclusions. Ibn Sīnā discusses an example in detail at *Qiyās* section 9.3, p. 442.8–443.13. The text is corrupt, but on one reconstruction Ibn Sīnā is discussing the separated syllogism with premises

- (16) ‘Every J is a D ’, ‘Every D is an H ’, ‘Every H is a Z ’, ‘Every Z is an I ’

and conclusion ‘Every J is an I ’. The intermediate conclusions are ‘potential’, he says. To find them, we start with two explicitly stated premises and draw a conclusion ϕ from them, and then we form a syllogism with ϕ as first premise and another of the explicit premises as second premise, and so on. An example would be to prove ‘Every J is an H ’ first, and then ‘Every J is a Z ’. He warns us against starting with the second and third premises to deduce ‘Every D is Z ’ — this is not ‘the arrangement that we chose’. (He adds that we could have chosen a different arrangement.)

Exactly this procedure, starting from the lefthand end, appears in Problem 3. (In Arabic of course it is the righthand end. I won’t say this again.) Ibn Sīnā takes a supposed separated syllogism of length 3 with the middle premise missing. He suggests a way of filling it, so that the three premises are

- (17) ‘No C is a B ’, ‘Every D is a B ’ and ‘Every A is a D ’.

He first infers ‘No C is a D ’ from the first two premises, and then he infers the required conclusion ‘No C is an A ’ from this and the third premise. Since this is one of the first problems, it’s presumably meant as a strong clue about the procedure to be followed.

So the procedure appears in the ASM of Appendix C as module (ASM4), called SYNTHESISE. Ibn Sīnā’s word for ‘synthesis’ is *tarkīb*, which means forming a compound; he also uses it for the compound formed. At *Qiyās* [15] p. 434.11 he explains that ‘synthesising a syllogism’ means forming a

connected compound syllogism, which is the main thing that this module does.

Now it's clear that if ϕ_1, \dots, ϕ_5 are formal sentences such that ϕ_1 and ϕ_2 entail ϕ_4 , and ϕ_3 and ϕ_4 entail ϕ_5 , then ϕ_1, ϕ_2, ϕ_3 together entail ϕ_5 . But Ibn Sīnā needs more than this. His procedure is also meant to tell us when the raw materials *can't* be filled out into a syllogism. Suppose we infer ϕ_4 from ϕ_1, ϕ_2 and then find that ϕ_5 doesn't follow from ϕ_3, ϕ_4 , what does this show? How do we know we couldn't have proved ϕ_5 from ϕ_1, ϕ_2, ϕ_3 by choosing ϕ_4 differently, or by starting at the righthand end? If Ibn Sīnā had tried to prove the correctness of his algorithm, he would have had to face this question.

In fact there is a positive answer, at least in terms of model-theoretic entailment. The heart of the matter is the following result.

Theorem 5 *Suppose $[\phi_1, \dots, \phi_n]$ and $[\psi_1, \dots, \psi_m]$ are linkages of formal sentences. Then the following are equivalent:*

- (a) $[\phi_1, \dots, \phi_n, \psi_1, \dots, \psi_m]$ forms an inconsistent minimal circle.
- (b) The set ψ_1, \dots, ψ_m has a strongest consequence θ , and $[\phi_1, \dots, \phi_n, \theta]$ is an inconsistent minimal circle.

The theorem tells us that (provided there are no irredundancies in the premises) we can take any segment of the premises of a separated syllogism, and shrink it down to its strongest consequence. The result will still be a separated syllogism entailing the same conclusion. At least this is true for model-theoretic entailment.

But consider for example the syllogism

- (18) Every B is a C . Every D is a B . Some D is an A . Therefore some C is an A .

Model-theoretically the three premises do entail the conclusion. But if we try to build a connected syllogism, starting from the lefthand end as in Ibn Sīnā's examples, we immediately hit a problem. The first two premises violate the fourth figure condition.

A possible way around this is to start by drawing a conclusion from the second and third premises. By the premise order condition this conclusion must be 'Some B is an A ', by the third-figure mood *Disamis*. So we have the intermediate syllogism

- (19) Every B is a C . Some B is an A . Therefore some C is an A .

This again is a valid instance of *Disamis*.

Hence Ibn Sīnā's procedure for constructing a connected syllogism from a separated one, in the form in which it appears in the Problems of his section 9.6, is inadequate. In fact he hits exactly this inadequacy at Problem 33. His solution is to switch the order of the first two premises in (18). The fact that he does this, rather than plough ahead with a different justification of the syllogism, is confirmation that he expects the student to start by drawing a conclusion from the two leftmost premises. This could be because he works from left to right, or because he shrinks the sequence of premises before filling the gap. Either way, this will fit our reading of the algorithm.

Until this glitch is sorted out, some doubt remains about exactly what separated syllogisms Ibn Sīnā would accept. Perhaps closer examination of *Qiyās* [15] section 9.3 will settle the point.

Theorem 5 merits a couple of further comments.

First, when we are trying to fill a gap in a sequence of premises, Theorem 5 tells us that if we can fill it at all without making any of the premises redundant, then we can fill it with a single sentence. Then Corollary 4 adds that there is a weakest single-sentence fill χ . When looking for linkages to fill the gap, we can confine ourselves to linkages that entail χ . It's not clear how far Ibn Sīnā was aware of this. For example at Problem 9 he notes that both 'Every D is a B ' and 'Some D is a B ' will fill the gap, but he fails to note that we have a better chance of finding a proof of 'Some D is a B ' (the weakest fill) than of 'Every D is a B '. (But as always, maybe he is encouraging his better students to see this point for themselves.)

The second comment is a technical warning. Suppose $m = 2$, ψ_1 has terms A, B and ψ_2 has terms B, C . Because of Ibn Sīnā's assumptions (6) about truth when a term is empty, θ in the theorem need not be logically equivalent to $\exists B(\psi_1 \wedge \psi_2)$. (Syllogisms don't have quantifier elimination.) So passing to θ might throw away information about A and C . But we can show that the lost information is recoverable from the rest of the circle.

4.5 Other kinds of syllogism

In [9.6.4] and [9.6.5] of *Qiyās* 9.6, Ibn Sīnā refers briefly to some other kinds of syllogism.

Earlier in the *Qiyās* ([15] p. 106) Ibn Sīnā has distinguished between two kinds of syllogism which he calls respectively 'recombinant' (*iqtirānī*) and 'duplicative' (*istitnā'ī*). A recombinant syllogism has two premises, each of them built out of two parts; one of these parts is the same in both premises.

The conclusion is formed by recombining the two remaining parts. Simple syllogisms as in subsection 4.3 above fit this description. But so do some propositional (*šartī*) syllogisms, for example

(20) If p then q . If q then r . Therefore if p then r .

Ibn Sīnā's view is that recombinant syllogisms are a generalisation of simple syllogisms, and that generally speaking the rules for simple syllogisms transfer to recombinant syllogisms too. (This is presumably what he has in mind at 468.7.)

Duplicative syllogisms are propositional. They have two premises. One of the two premises has two parts. The other premise consists of one of these two parts (or its contradictory), and the conclusion consists of the other part (or its contradictory). The shorter premise and the conclusion are said to be 'duplications' (i.e. of parts of the longer premise). Besides modus ponens:

(21) If p then q . p . Therefore q .

this description covers inferences like:

(22) Not both p and q . p . Therefore not q .

Ibn Sīnā regards duplicative syllogisms as incomplete in themselves; they only make sense as part of a longer argument. There seems to be no natural way of generalising his proof search procedure to them.

Ibn Sīnā classifies binary sentence connectives and the compounds that are formed using them as 'meet-like' (*muttaṣil*) or 'difference-like' (*munfaṣil*). This is a soft classification based on some supposed resemblance to meet ('and') or difference (exclusive 'or'). But he doesn't use the classification consistently, and my present impression is that he never settled on a satisfactory principle for classifying binary sentence connectives. The 'If . . . then' in (21) counts as meet-like, while 'Not both' in (22) counts as difference-like; so these two syllogisms are respectively meet-like duplicative and difference-like duplicative.

There is more on Ibn Sīnā's propositional syllogisms in Shehaby [29], together with a translation of the propositional part of *Qiyās*. Shehaby translates the technical terms differently: he has 'conjunctive' for 'recombinant', 'exceptive' for 'duplicative', 'conditional' for 'propositional', 'connective' for 'meet-like' and 'separative' for 'difference-like'.

5 Extracting the algorithm

Can we be sure that Ibn Sīnā really meant to describe an algorithm for proof search?

Ibn Sīnā himself doesn't say anything to indicate that he regards the procedure that he is teaching as comparable with the kinds of algorithm known to medieval Arabic mathematicians (see subsection 6.2 below). He does say in [9.6.1] that we need 'rules' (*qawānīn*, plural of *qānūn*) to guide us in analysis. But in [9.6.2] he explains this as 'rules in the form of dos and don'ts', which doesn't sound like an algorithm. At *Qiyās* [15] p. 537.3 he uses the same phrase 'dos and don'ts' for advice about how to conduct a debate.

In fact the passage that I interpret as describing an algorithm (paragraphs [9.6.6] to [9.6.11]) consists of 64 problems, with answers given and some remarks about how the answers are found. The problems are divided into four groups according to their patterns; Ibn Sīnā explains the patterns and tells the reader 'Do the remaining cases of this pattern for yourself' (463.12, 464.12, 466.2, 467.7). The problems are introduced without any explanation of what they are for, apart from the fact that they appear in a discussion of analysis. At the end of them Ibn Sīnā comments (468.4f):

- (23) When you put the steps in this order, as I have shown you, you will reach the terms, figures and moods.

Strangely the 64 problems make no mention at all of syllogistic moods, and only one mention of figures. But figures and moods are a classification that makes sense only for simple syllogisms, so Ibn Sīnā is implying here that by 'putting the steps' in the right order we will be able to reduce the raw material in the problems to simple syllogisms. Since no new kinds of 'step' are described here, it seems to follow that Ibn Sīnā means we can use steps already discussed earlier in *Qiyās* to reduce the raw material to simple syllogisms. This exactly matches the use of the module SYNTHESISE in our ASM, which rests on procedures described in *Qiyās* section 9.3. So it confirms that we are on the right track.

In any event, paragraphs [9.6.6] to [9.6.11] are clearly intended to teach the reader a procedure for taking data of a certain kind and coming up with answers to certain questions about the data. The decision whether to call this procedure an algorithm is for us, not for Ibn Sīnā. Our choice should rest on three issues: (1) Is the class of data to which the procedure applies well-defined? (2) Is it clear what question or questions the procedure is meant to answer? (3) Is the procedure mechanical?

I take these issues in turn. Of course the procedure defined by the ASM in Appendix C is an algorithm, but we need to ask how much of that algorithm is already in Ibn Sīnā's text.

5.1 The class of input data

The 64 problems all share a common format. They involve syllogistic sentences with letters for terms; but we are not told what the letters stand for. So there is no loss in thinking of the sentences as formal sentences, provided that we don't impose our definition of entailment on Ibn Sīnā.

Each problem begins with a syllogistic sentence called the 'goal' (*maṭlūb*), except where we have to understand that the goal is the same as in the previous problem. Then follows a sequence of one or more formal sentences. Ibn Sīnā's commonest description for this sequence is that 'you have' (*kāna ʿindak*) the sentences in it; he uses this or closely similar phrases in 38 problems. The only name that he offers for the sequence is 'thing found' (*mawjūd*, in 8 problems). This word has a variety of meanings in Ibn Sīnā's logic, and another variety in his metaphysics. But since he also says in 6 problems that 'you have found' (*wajadta*) the sequence, I assume it just means 'what has been found'. For the sake of English style I shorten this to 'datum'; I follow Ibn Sīnā's lead in using the singular even when there is more than one sentence in the sequence.

So each problem has a goal and a datum. In every case but one, the sequence consisting of the datum followed by the goal is a linkage in the sense of subsection 4.2 above. The one exception is Problem 33, where the linkage order would run foul of the fourth figure condition (see the note on the Problem). This exception shows that Ibn Sīnā does expect the student to be able to handle a larger class of inputs than the ASM in Appendix C is designed for. But I think we can regard this Problem as a freak case.

Ibn Sīnā indicates at 464.12f, 465.5, 466.6, 467.7 and 467.10 that we should think of the datum as having two parts, one that has a sentence containing the subject of the goal and one with a sentence containing the predicate of the goal; except that one of these two parts may be empty. What he describes here is exactly the gap site that we calculated in subsection 4.2; the part of the datum linked to the subject is precisely the part before the gap indicated by the gap site.

It's curious that Ibn Sīnā explains this structure of the datum only after the 26th problem. Perhaps some text has gone missing, but I doubt it. He has a tendency to explain himself only after he has given you a chance to work out for yourself what he must have meant. My impression is that the

Arabic mathematicians of his time would have regarded this as poor style.

All four kinds of formal sentence appear in Ibn Sīnā's problems, in a wide variety of combinations. So it seems that the class of properly ordered possible goal-datum pairs is well defined, except perhaps for the questions of length and of the number of gaps. To begin with length, in all Ibn Sīnā's examples the datum has length 1 or 2. Did he intend his procedure to apply only in these cases?

I believe not, for two reasons. The first is that in Problems 1 and 2 he points out that if we can't find a suitable single sentence to fill the gap, we may need to look for a pair. He doesn't say what happens next, but one reasonable way forward would be to guess (say) the first sentence of the pair and put it into the datum. Finding the second sentence would then be the original problem but with a longer datum. (And so on recursively, though he never says this.)

Strictly this is not the only way forward. As we will see, the question of looking for a pair of sentences only arises after we have discovered a weakest fill ϕ for the gap in the original datum. Then by Theorem 5 above, it suffices to continue with ϕ as new goal and an empty datum. But even this would add 0 to the possible lengths of data. I haven't followed this route, because it would imply some mechanism for feeding back the result of the calculation with ϕ as goal into the original problem.

But the case of length 0 is interesting anyway, not least because it corresponds to the Prolog proof search problem. For that reason I set up the ASM to handle data of length 0. Ibn Sīnā himself may have reckoned that he had said enough about the case of data of length 0 already in *Qiyās* [15] section 9.4 'On obtaining premises, and on *tahṣīl* of syllogisms with a given goal'.

The second reason for doubting that Ibn Sīnā intends a restriction to lengths 1 and 2 is his statement at 465.2 that he will deal with the case of 'more than two premises' in the appendices. We don't have the promised appendix; see the Note on this passage. Of course he might have said in the appendix that these longer data can be handled, but only by a different procedure. I think this is unlikely, for the first reason just given.

Nevertheless there is a good reason for Ibn Sīnā to concentrate on the case of length ≤ 2 . If the datum has length greater than 2, it always contains two adjacent sentences that share a term. So we can reduce the length of the datum immediately, by replacing these two sentences by their strongest consequence — unless they are sterile, in which case the problem has no positive solution. We can't be sure that Ibn Sīnā intended this way of working, but it makes good sense and I have built it into the ASM.

The other possibility is that Ibn Sīnā intends his procedure to apply where the datum contains more than one gap, or perhaps even when it contains no gap at all. He does in fact discuss the case of more than one gap in paragraph [9.6.7]. His view is that it can be handled but at the cost of a more complicated procedure, which again he will describe in the appendices. The main thing we would need to do in order to extend our ASM to more than one gap would be to incorporate some further machinery to control the search; see subsection 6.2 below for a discussion of what would be required. Presumably Ibn Sīnā’s appendix would have said something about this too. The case of no gaps is covered by the procedures of *Qiyās* section 9.3, which we have incorporated into the module SYNTHESISE; so this case is at least implicitly in Ibn Sīnā’s algorithm already.

In his initial remarks on analysis in [9.6.1], Ibn Sīnā says that the text to be analysed may contain ‘something superfluous’, and our rule will need to tell us how to ‘strip off defects’. This suggests that the procedure should also eliminate redundant parts of the datum. None of the 64 problems suggests any way of doing this. Indeed it’s not clear what the aim would be if Ibn Sīnā did allow this. One could always start by removing the entire datum and working from the goal alone; would this count? If not, would the aim be to throw away as little as possible of the datum? This could lead to serious complexities. So I think we can sensibly assume that the procedure is not meant to eliminate redundant parts of the datum.

5.2 What question is answered?

Alongside each one of his 64 problems, Ibn Sīnā provides an answer. With trivial variations, all the answers take one of two forms. The affirmative form is: If the sentence χ is attached (*ittasal*, 27 problems) then ‘it has been made *ḥāṣil*’ (*qad ḥuṣṣil*, 17 problems). Usually Ibn Sīnā doesn’t tell us what has been made *ḥāṣil*. But at Problems 8 and 9 he does: it’s the syllogism. (See also Problem 1: ‘your syllogism is in good order’.) The translation below reflects this.

By ‘attached’ he clearly means ‘put into the gap in the datum’. So the procedure involves an operation that does this. I was tempted to call this operation ‘attach’, but unfortunately this is a reserved word in the vocabulary of ASMs. Since the operation is a syntactic triviality, I made it not a module but a basic function: (Def5) in Appendix C.

The negative form of answer is: ‘It can’t be used’ (*lam yuntafa^c bih*, 23 problems). This time we can hardly expand to ‘The syllogism can’t be used’, because in these problems there is no syllogism. A more accurate

expansion would be ‘The goal and datum can’t be used to generate a *ḥāṣil* syllogism’; but for brevity I stick with ‘it’ in the translation.

The problems with a negative answer are exactly those in which there is no sentence that can be put in the gap of the datum so as to yield a separated syllogism with the goal as conclusion. Also in the problems with an affirmative answer Ibn Sīnā nearly always names a sentence that can be put in the gap so as to yield the required syllogism. So this is at least one of the aims of the procedure:

- (24) *Determine whether or not there is a sentence χ that can be put into the gap of the datum so that the datum becomes the premise sequence of a separated syllogism whose conclusion is the goal. When the answer is Yes, supply a sentence χ with this property.*

I call this the ‘logical task’. Note that it makes no reference at all to sentences that are already *ḥāṣil*.

Note also that the logical task, as stated, doesn’t include classifying the resulting syllogism by means of figures and moods. We saw earlier that in fact Ibn Sīnā’s procedure, if we have reconstructed it correctly, does yield enough information to convert the separated syllogism into a connected one, and then the figures and moods can be read off. So we could add a further module to the ASM which delivers the connected syllogism with its simple syllogisms labelled by figure and mood. But this would mean introducing a new datatype for connected syllogisms, and it would be just a unit bolted onto what is already in the ASM. So I take a lead from Ibn Sīnā, who mentions in [9.6.12] that the procedure will yield this information but gives no further details. I add only that one possible implementation of the ASM is in terms of diagrams written on paper, very likely as Ibn Sīnā’s students would have drawn them. These diagrams would almost certainly have included the connected syllogisms.

But Ibn Sīnā also says a number of other things that only make sense if he is expecting the procedure to deliver a syllogism that is *ḥāṣil* in the sense we studied in section 3 above. First and foremost, there is the wording that we quoted in the affirmative case: ‘[the syllogism] has been made *ḥāṣil*’. Add to this that in 10 problems he says that the premises in the datum are *ḥāṣil*; this is irrelevant for the logical task. In 6 of the problems with an affirmative answer, he requires that the attached sentence is ‘true’ or ‘true for you’ or ‘clear’ (*bayyin* — this must mean ‘clearly true’). Finally there are two problems (1 and 2) where Ibn Sīnā finds a sentence χ that solves the logical task, and then adds that if the sentence is not ‘clear’ or true, then it

doesn't solve the problem and one 'needs a middle' (i.e. has to look for a two-sentence filling for the gap).

So there is clear evidence that Ibn Sīnā also has in mind another task:

- (25) *Given that the datum consists of sentences that are already ḥāṣil, determine whether or not there is a sequence of sentences $[\chi_1, \dots, \chi_m]$ that are already ḥāṣil, which can be put into the gap of the datum so that the datum becomes the premise sequence of a ḥāṣil separated syllogism whose conclusion is the goal. When the answer is Yes, supply a sequence $[\chi_1, \dots, \chi_m]$ with this property.*

I call this the '*taḥṣīl* task'. The two tasks are connected by the fact that a negative answer to the logical task implies a negative answer to the *taḥṣīl* task, but otherwise the tasks are independent.

I think it's inconceivable that Ibn Sīnā was in any way confused about the difference between the logical task and the *taḥṣīl* task. But I wouldn't put it past him to be deliberately ambiguous in hopes of catching both tasks under the same general description. There is some evidence of deliberate ambiguity. In subsection 5.1 we interpreted the word 'found' (*mawjūd*) as meaning datum, i.e. 'the thing you found in front of you when you were given the problem'; but it would be entirely in keeping with Ibn Sīnā's logical vocabulary if we read it as 'found to be true', i.e. *ḥāṣil*. Likewise the phrase 'you have' (*kāna 'indak*) could also mean 'according to you', in other words, 'it's *ḥāṣil* for you that ...'.

It would also be in character for Ibn Sīnā to leave the ambiguity as a deliberate trap for idle or unintelligent students.

In sum, we have identified two tasks that the procedure is meant to perform. The logical task is well-defined apart from the uncertainty about what separated syllogisms Ibn Sīnā accepts. But at least we can rigorously check the correctness of Ibn Sīnā's own solutions of his 64 problems. The *ḥāṣil* task is well-defined apart from the same uncertainty about separated syllogisms, though it does require us to know what sentences are 'already *ḥāṣil*'. The set of things that are already *ḥāṣil* is the counterpart of the set of clauses of the Prolog program in the Prolog case. Börger and Rosenzweig [5] build this set of clauses into their ASM through a predicate *PROGRAM* and a basic operation *clause_list*. I prefer not to do that here, because it would pre-empt a question we have to discuss in a moment, namely whether Ibn Sīnā considers that the set of sentences that are already *ḥāṣil* can be read off mechanically.

5.3 Is the procedure mechanical?

Ibn Sīnā doesn't ringfence his procedure; we have some discretion to decide what counts as part of it and what involves an appeal to the environment. The real question here is whether the procedure has a purely mechanical core, and if so, what that core contains.

Most of the procedure is quite obviously mechanical. Although Ibn Sīnā refers at [9.6.12] to 'putting the steps in this order', he is a little vague about what that order is. But as far as I can see, the indeterminacies are all of the kind where it doesn't matter what order we choose, and it's routine to find a mechanical arrangement of the steps that does the required job.

There are three places where Ibn Sīnā relies on the reader to have a certain skill. The first is the computation of the strongest consequence of a non-sterile pair of premises. This is the job of the basic function *consequence* at (Def3) in Appendix C. If the worst comes to the worst, the function can be implemented by simply listing the possible cases, as in an appeal to the student's memory.

The second place is where, in the *taḥṣīl* task, Ibn Sīnā asks the student whether a certain named sentence is already *ḥāṣil*. I see no problem about taking this as a basic function *hasil* of the ASM, as in (Def6) in Appendix C. It doesn't necessarily follow that the sentences that are already *ḥāṣil* can be listed by listing all sentences and then filtering through the function *hasil*, because the set *SENTENCE* could be dynamic. More precisely there could be infinitely many sentences, or more than are listed in the set *SENTENCE* in the ASM at the outset of the computation, and the ASM may be able to add further sentences *SENTENCE* as the computation proceeds. (Since *SENTENCE* is defined in terms of *TERM*, this would involve adding new terms to *TERM* too.) This possibility doesn't arise when a single given sentence is being evaluated for being *ḥāṣil*.

The third place where the reader needs a skill is where Ibn Sīnā says (in Problems 1 and 2) 'it needs a middle'. The situation is that a sentence χ has been identified as the weakest fill for a certain datum, and the function *hasil* has been used to reveal that χ is not already *ḥāṣil*. I have to mention another glitch hidden here. It could happen that χ is not itself already *ḥāṣil*, but it is a consequence of a one-premise inference (as in (8) from a premise θ that is already *ḥāṣil*. The algorithm should identify θ and put it in place of χ . This needs an extra piece of machinery which I haven't included in the ASM. One excuse I can offer is that putting θ in place of χ could possibly lead to a violation of the fourth figure condition, and we don't know what Ibn Sīnā thinks about this possibility.

In any case, the statement ‘it needs a middle’ is shorthand for:

We need to look for a term C and sentences ϕ_1, ϕ_2 using the terms of χ and the term C , so that ϕ_1, ϕ_2 are already *ḥāṣil* and are the premises of a syllogism with conclusion χ and C as middle.

Ibn Sīnā discusses this situation in a number of places.

For example *Qiyās* [15] section 9.4 is about this question. Ibn Sīnā advises that we start by looking at the form of χ . Thus suppose it has the form ‘Every A is a B ’. Then we should unpack the definition of the term A , and extract from it sentences of the form ‘Every A is a C ’. For each of these, we should see whether we can also prove ‘Every C is a B ’. If we have no success with the definition of A , Ibn Sīnā advises looking next at the properties that we can prove for A , using the principles of the relevant science.

In the cases where χ has the form ‘No A is a B ’ or ‘Some A is a B ’, the situation is symmetrical and we can start with either A or B . In the case of ‘Some A is not a B ’, Ibn Sīnā’s wording suggests — I can’t put it stronger than that — that we start with properties that some A is known to have. So a general rule that covers all cases would be that we start by looking for *ḥāṣil* sentences that involve the subject term of χ . (Note that the subject term could be either the left edge or the right edge of the gap.)

Ibn Sīnā comes back to the matter at *Burhān* [16] pp. 138.22ff and 139.10ff. He claims that in mathematics most sentences have the form ‘Every A is a B ’ (here he is agreeing with Aristotle *Posterior Analytics* A14). He suggests that when χ has this form in mathematics, if there is a middle as required, then one can be found by unpacking the definition of the subject term of χ . (This seems to me a gross oversimplification outside elementary linear algebra.) In this case it would be reasonable to say that the list of possible terms can be found mechanically from the definition of the subject term, so we would only need to include in the ASM a basic function for finding the definitions of terms. But Ibn Sīnā goes on to say that outside mathematics things are not so straightforward. We would need to consider the inherent accidents of the subject term of χ , and in the worst case even its non-inherent accidents.

He comes back again to the same question in his autobiography. He tells us that sometimes he was ‘at a loss about a problem, concerning which I was unable to find the middle term in a syllogism’, and so he resorted to prayer, then to alcohol and then to sleep; ‘many problems became clear to me while asleep’ ([12] p. 27f). Prayer, alcohol and sleep are not mechanical procedures.

All in all, I think it would be very unwise to assume that Ibn Sīnā thinks we can list in advance all the *ḥāṣil* sentences that involve the subject term of χ . This is a pity, because the backtracking algorithm of [5] (which Börger and Stärk display as an ASM module on page 114 of [6]) assumes that we can make this list.

At this point I am going to cheat and call on a relatively advanced kind of Abstract State Machine called an asynchronous multi-agent ASM ([6] Chapter 6). This multi-agent ASM has a family of ‘agents’ who each perform according to their own ASMs, at their own speeds and for the most part independently. But there can be super-global procedures that pass messages to and from the agents. The set of agents can be ‘potentially dynamic’, in other words there can be super-global procedures that add new agents. In ASMs one can treat the set of threads in a Java program as a dynamic set of agents; I thank Egon Börger for this example. (The term ‘super-global’ is to distinguish from those features of the agent ASMs that are global within these ASMs.)

In this setting, suppose an agent reaches a point where ‘it needs a middle’. The agent then sends a message to the super-global agent who operates the super-global procedures; prayer, alcohol and sleep might be ways of sending this message. The super-global agent responds by listing all the possible options; but instead of sending the list to the agent, it splits the agent into a family of agents, each of whom has one of the options to work on. I see Ibn Sīnā identifying the global agent as the Active Intellect, and the agents who carry out the algorithm as possible intellects, so that

- (26) when a connection occurs between our souls and [the Active Intellect], there are imprinted from it in them the intellected forms which are specific for this specific preparation for specific judgments. (*Iṣārāt* [20] II.3 iṣ. 13.)

But that’s an aside — the super-global agent has a precise job to do, which is encoded in the ASM as a super-global basic function.

All the agents do the same calculation for the logical task. When the logical task has delivered an affirmative answer, they switch to the *taḥṣīl* task and may have to split. So for the *taḥṣīl* task we need to clarify the notion of correctness of the ASM, as follows. *The ASM is correct for the taḥṣīl task if: (1) when the task has a negative answer, all (lower level) agents return a negative answer; (2) when the task has an affirmative answer, at least one agent returns an affirmative answer; and (3) every agent returning an affirmative answer also returns a sequence of sentences which is a correct fill for the gap in the datum.*

A fragment of the backtracking procedure is still needed, but for a more limited purpose, namely to find the weakest fill in a datum. Ibn Sīnā shows at Problems 3 and 7 that he expects the student to find it by listing possibilities and trying each in turn. The edges of the gap are known, and they provide the two terms of the weakest fill. So there are eight possible sentences to consider. Given this approach, it makes sense to list the possibilities in an order where ψ comes before χ whenever χ entails ψ ; so when we first find a possible fill we know it is a weakest one. The function *listsentences* at (Def2) in Appendix C provides such a list.

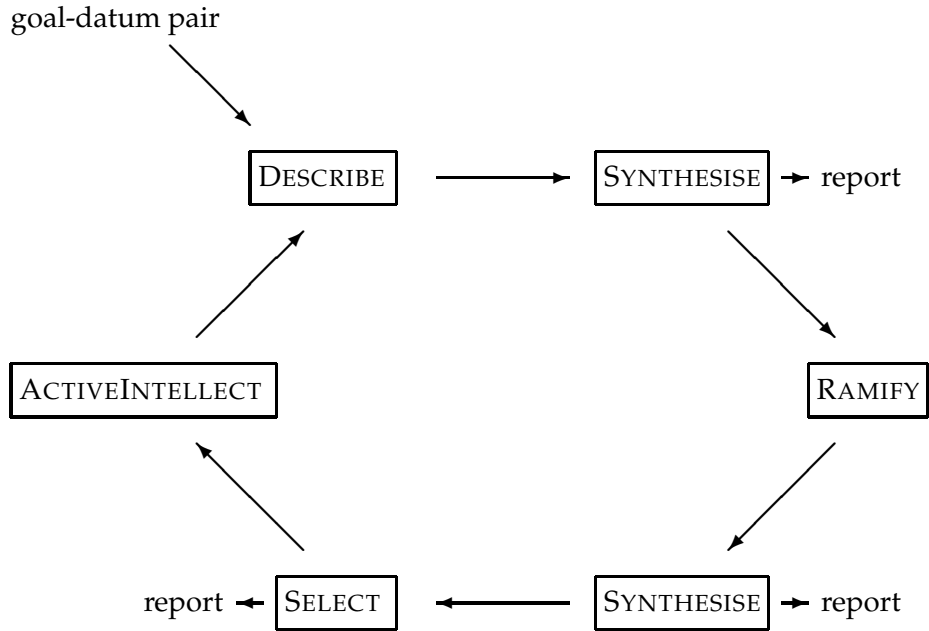
Ibn Sīnā allows the student to use background knowledge to cut down from eight to a shorter list of possible fills; see the notes on Problems 3 and 7. I count this move as a shortcut, not as a part of the algorithm.

Are we sure that no further backtracking is needed? For example, perhaps we find a weakest fill, but then further down the line we discover that the resulting connected syllogism runs into trouble with the fourth figure condition, so that we need to backtrack and try the converse of the weakest fill instead. I believe that this problem doesn't arise, because the premise order condition fixes the order of the terms in all the intermediate sentences in the connected syllogism, independent of the order of the terms in the premises of the separated syllogism. To be sure of this we need a correctness proof; but I think this would be wasted effort until we have an answer to the question about which connected syllogisms to accept.

6 Review

We must do two things here. The first is to give an informal summary of the algorithm, and the second is to place it in the history of logic and mathematics. A more formal description of the algorithm is given in Appendix C, in the form of an asynchronous multi-agent ASM, where each agent follows its own agent ASM within the multi-agent ASM.

6.1 Summary of the algorithm



We describe what happens to a goal-datum pair as it proceeds through the diagram above.

Entering the module `DESCRIBE`, the goal-datum pair is measured up to determine where its gap site is and what the left and right edges of the gap are. This information is attached to it for future use.

Then it proceeds to the module `SYNTHESISE`, which shrinks it down. If there is a pair of adjacent sentences in the datum that have a term in common, `SYNTHESISE` works out the strongest consequence of these two sentences, and replaces the sentences by this strongest consequence. It does this starting with the leftmost such pair of sentences, and continues until either there are no such pairs left, or it reaches a pair that is sterile. In the latter case it reports failure and the algorithm halts.

If the shrunken goal-datum pair survives through `SYNTHESISE`, it passes to the module `RAMIFY`. This module determines the eight sentences ϕ_1, \dots, ϕ_8 whose terms are the two edges of the gap. The sentences are listed so that if ϕ_i entails ϕ_j but not vice versa, then $j < i$. Then the module splits the goal-datum into eight clones, and it fills the gap in the i -th clone with the sentence ϕ_i . So now there are eight goal-datum pairs, none of which has a

gap.

There is a subtlety if the goal-datum pair that passes to RAMIFY has an empty datum. In this case there is always a sentence that fills the gap and entails the goal, namely the goal itself. So in this case RAMIFY makes just one new page, in which the datum is changed to the goal sentence.

After RAMIFY has done its work, the first of the resulting gap-free goal-datum pairs passes to SYNTHESISE, which shrinks down any adjacent pair of sentences in the datum with a term in common, until either the datum consists of a single sentence, or a sterile pair of sentences has come to light. If a sterile pair of sentences comes to light, the goal-datum pair is discarded and the next of the eight clones passes into SYNTHESISE for similar treatment; and so on. If none of the eight clones are left, the module reports failure.

If a goal-datum pair with a single-sentence datum survives, it passes to the module SELECT. This module checks which of three cases hold: (1) the datum equals the goal, and it is already *ḥāṣil*; (2) the datum equals the goal, but it is not already *ḥāṣil*; (3) the datum doesn't equal the goal. In case (1) the module reports success in the *taḥṣīl* task and the algorithm halts. In case (2) the module reports success in the logical task (if it hasn't already been reported), restores the gappy goal-datum pair that RAMIFY had filled, and sends this pair to the Active Intellect with a request for a *ḥāṣil* sentence that attaches at one side of the restored gap. The Active Intellect compiles a list of all the *ḥāṣil* sentences that could be used, and it makes one clone of the goal-datum pair for each such sentence ψ . The clone that goes with ψ has ψ inserted into its gap; but the gap is not completely filled, so we once again have a goal-datum pair with a gap. All these new goal-datum pairs are sent back into DESCRIBE in parallel, and so on around the cycle. In case (3) the same happens as the failure case in the previous paragraph: the goal-datum pair is discarded and the next of the eight is called for, unless none of the eight are left, in which case the module reports failure.

There are several places where a module reports success or failure. If no success has been reported yet, then the first report of success or failure is a report on the logical task, except in case (1) for SELECT. If logical failure has been reported, the algorithm halts. If logical success has been reported, a later report of failure is a report on the *taḥṣīl* task, and again the algorithm halts. If logical success has been reported, the only further report of success that makes any difference is a report of *taḥṣīl* success in case (1) for SELECT.

This is the algorithm in broad outline. We need to clarify what are the separate steps, and how the algorithm determines which step happens

when — what Ibn Sinā refers to as the ‘order’. The description below is very much based on Gurevich’s notion of an ASM and the use made of it by Börger and Rosenzweig in [5].

The idea of goal-datum pairs swimming around between modules is only a metaphor. A different metaphor is more realistic: the calculator (or ‘agent’) does each piece of calculation by writing out one or more pages that state the results of the calculation. (The pages are the ‘nodes’ of [5].) A step of the calculation could involve writing several pages, but only where the pages can be written simultaneously. For example when RAMIFY makes eight clones and fills them, in principle this can be done on eight pages simultaneously (though eight hands would be useful), so it counts as a single step. But when SYNTHESISE shrinks down the datum, the result of shrinking down the first pair of sentences is generally an input to the operation of shrinking the next pair. So shrinking down a single pair of sentences to their strongest consequence is a whole step. In general SYNTHESISE will process a goal-datum pair for several steps until there is no fat left on the datum; this will involve producing a succession of new pages with shorter datum sequences.

In principle the agent could go to work on any existing page at any time, using any one of the four modules DESCRIBE, SYNTHESISE, RAMIFY or SELECT. What determines which page and which module the agent will take next?

Written in a separate place, not on the pages, there are three further pieces of information stored in ‘global variables’. The first is the label of the ‘current page’, i.e. the page now being processed. The agent reads the current page and acts according to instructions in the algorithm; these instructions refer to the contents of the current page, and to the values of the global variables. The instructions tell the agent what new pages to produce, and what changes to make to the global variables. So for example if the agent is looking at page 5, the instructions may tell the agent to change the current page variable to 6; the effect is that when page 5 has been dealt with, the agent turns next to page 6. And so on.

There are two other global variables besides ‘current page’. One of them records the goal (which is fixed at the start and never changes). The other global variable stores reports of success or failure (and starts with the value ‘ignorance’).

The rest of the information needed for controlling the calculations consists of six records on each page, as follows. (In Appendix C these six records are called ‘properties’ of the page.) The first is a record of the datum on that page. (The starting page carries the datum given by the problem to

be solved.) The second records the gap site for the current goal-datum pair; the record may also show that there is no gap, or that the gap site needs calculating. The third is a record of the left and right edges of the gap. The fourth is a record of the fill, i.e. the sentence that was put in the gap when RAMIFY was last used.

The fifth and sixth records on the page store information about the movement between the pages. One of them records 'previous page'; what this means is that when a page p is being read and a new page q is constructed according to the information in p , then p is recorded as 'previous page' on q . (After the algorithm has reported success, one will need to work backwards from the final page to its previous page, its previous page's previous page and so on in order to reconstruct the required connected syllogism.) The other record is called 'next'. The main function of 'next' is that when a group of pages p_1, \dots, p_n are constructed simultaneously, 'next' on page p_i (where $i < n$) indicates p_{i+1} . When the agent is reading p_i and has to discard it, the value of 'next' on p_i tells the agent which page to try next. The agent makes this happen by changing the value of the global variable 'current page' to p_{i+1} .

If we have the algorithm set up correctly, then at any stage the records on the current page p will determine uniquely which module takes care of this stage of the calculation — unless the algorithm has halted with a report of success or failure. For example if the record of the gap site on p says that there is no gap, the module that applies will be one of the two at the bottom of the diagram. If and only if the record says that the gap needs calculating, the module that applies will be DESCRIBE. If the record says that there is a gap, the module that applies will be one of SYNTHESISE, RAMIFY and ACTIVEINTELLECT.

The module ACTIVEINTELLECT, which is operated by a higher force, comes into play when and only when the record 'next' on the current page indicates prayer, alcohol or sleep — or more prosaically when it has the value 'needs a middle'.

Assuming that neither DESCRIBE nor ACTIVEINTELLECT has been called, what settles the choice between SYNTHESISE, RAMIFY and SELECT? The answer is that SYNTHESISE applies if and only if the datum on the current page has two adjacent sentences with a term in common. (The module appears twice in the flow diagram above, but it has the same job to perform in both cases.) If SYNTHESISE doesn't apply, then RAMIFY applies if there is a gap in the goal-datum pair, and SELECT applies if there isn't.

It may be helpful to note that when RAMIFY or SELECT applies to a page,

then the datum on the page has been slimmed down as much as possible by SYNTHESISE. So if the goal-datum pair has a gap (as at RAMIFY), the datum consists of at most two sentences; if the pair has no gap (as at SELECT), the datum is a single sentence. I suggested earlier that this explains why Ibn Sīnā confines his 64 problems to cases where the datum has at most two sentences.

For further details refer to Appendix C.

6.2 The place of the algorithm in history

My remarks here will be very incomplete — this is already a long paper. I am very much indebted to Roshdi Rashed for his comments and information, though he should not be held responsible for any particular claims I make.

A ‘search algorithm’ is a mechanical procedure which allows its user to find a solution of a problem, or establish that there is no solution, by running systematically through a set of possible partial or total solutions. (The set is called the ‘search space’.) Ibn Sīnā’s algorithm, insofar as it really is an algorithm, is a search algorithm for finding solutions to the logical and *tahṣīl* problems. It searches through partial or total compound syllogisms that extend the datum. I know of no other examples of search algorithms in the medieval Arabic literature. In modern times search algorithms go back at least to Tarry’s maze-solving algorithm of 1895 ([4] p. 18ff), though the best known examples are from the second half of the 20th century.

Closely related to search algorithms are two other kinds of algorithm. A ‘counting algorithm’ allows its user to calculate the number of elements of a given set. A ‘listing algorithm’ allows its user to list without repetition all and only the elements of a given set. Search algorithms sometimes use listing algorithms to list the elements of the search space; but unless the listing is appropriate for the problem, the resulting search algorithm can be very inefficient. Sometimes a counting algorithm can be proved correct by examining a listing algorithm. For example one can show that the number of elements of the cartesian product $X \times Y$ is the product of the number of elements of X and the number of elements of Y by examining the lexicographic product of X and Y (as done for example by Ibn Mun’im in the 13th century, Katz [22]).

The earliest reported algorithms in Arabic mathematics are listing and counting algorithms in connection with strings of letters. They appear first in the *Kitāb al-ʿAyn*, an 8th century linguistic text normally attributed to the polymath al-Ḳalīl ibn Aḥmad (though the situation must be more compli-

cated, because the book includes third-party reports of al-Ḳalīl’s views, cf. Versteegh [32] Chapter 2). The basic problem that al-Ḳalīl addresses is how to list the words of Arabic in a dictionary. His preferred ordering is not lexicographic; rather he lists unordered sets of consonants, and for each set he makes a sublist of its permutations. He introduces this ordering with some combinatorial calculations that count numbers of permutations. In the 9th century Ibn Duraid developed these calculations, describing them as a sort of calculus (*ḥisāb*). Cf. Rashed [27] p. 18ff for this aspect of the *Kitāb al-ʿAyn* and its later influence.

In the 9th century Muḥammad ibn Mūsā al-Ḳwārizmī introduced the classical algorithm for solving quadratic equations. More than this, he demonstrated the ‘cause’ (*ʿilla*) of the algorithm; in other words, he gave a mathematical demonstration that the algorithm always yields a correct answer when the coefficients of the equations are real numbers (or less anachronistically, when they can be represented as lengths). For this he converted from numbers to lengths, and then invoked geometrical arguments in the style of Euclid. Details are in [27]. In the rich tradition inspired by al-Ḳwārizmī’s work, the word for ‘algebraic algorithm’ is *bāb*, translated into Latin as *regula*.

None of the algorithms of al-Ḳalīl or al-Ḳwārizmī are search algorithms, and Ibn Sīnā gives no indication that he sees himself as doing anything similar to what they did. He never describes his procedure as a *ḥisāb* or a *bāb*. There is some overlap with al-Ḳalīl’s calculations, namely that Ibn Sīnā uses lists of possibilities. But unlike al-Ḳalīl, Ibn Sīnā spends no time discussing systematic ways of listing the possibilities. Ibn Sīnā would have had to consider some kind of backtracking algorithm if he had taken more seriously the implications of the *taḥṣīl* problem; but this would have moved him into territory unknown to any medieval mathematician (as far as we know).

To adapt his algorithm to problems with more than one gap, Ibn Sīnā would have had to search systematically through the cartesian product of the sets of possible fills at the separate gaps. If he proposed to do this by listing the possibilities lexicographically — and it’s hard to think of any other reasonable procedure — then this would have brought him close to al-Ḳalīl’s listing procedures, and he would very likely have given us the earliest description of a search through lexicographic listing. This makes it all the more painful that we don’t have the appendix which he said would discuss problems with more than one gap. (See the note on 465.2.)

There is a major difference between Ibn Sīnā’s discussion and that of al-Ḳwārizmī. Namely, Ibn Sīnā never makes any attempt to show that his

algorithm is correct. (If he had done, he would certainly have given a much better algorithm.) There are several aspects to this difference. First, al-Ḳwārizmī is following every mathematician's dream: to solve a problem by reducing it to some apparently quite different problem that is easy to solve or has already been solved. The reduction of a problem in algebra to one in geometry is a beautiful example; and incidentally it runs clean counter to the aristotelian tendency to keep the various sciences in a rigid hierarchy. I doubt that Ibn Sīnā ever had this mathematical dream. In the same way as Aristotle, he writes mathematics like an intelligent outsider, not like a true addict.

The second aspect is about how Ibn Sīnā sees the nature of logic. For Ibn Sīnā logic is not about when this follows from that; it's about *how we can see from first principles that this follows from that*. For example if we are given a linkage and a sentence, Theorem 2 gives a fast way of testing whether the linkage entails the sentence without needing to construct any simple syllogisms at all. In Ibn Sīnā's time this theorem wasn't yet known. But even if it had been, it would have established a logical fact by going outside the basic processes of deduction, and so Ibn Sīnā very probably wouldn't have used it. The fact that Ibn Sīnā uses only direct and bottom-level methods was a great help for extracting the algorithm from Ibn Sīnā's text. One knew in advance that there were no hidden tricks or changes of viewpoint or appeals to intuition. The student was expected to solve the problems by direct application of basic facts of logic, and all that Ibn Sīnā was teaching him was how to apply the steps in the right order (as he himself says at [9.6.12]).

For balance one should add that in general Ibn Sīnā was certainly prepared to use metatheorems of logic as well as theorems. In fact he despised logicians who couldn't do this. But the metatheorems that he used were ones that summed up elementary facts about syllogisms, not ones that introduced new ideas.

In one other respect Ibn Sīnā's algorithm matches the mathematics of his time. He achieves the effect induction by reducing more complex cases to simpler ones, until he reaches ground level. We might compare Proposition 8 of Ṭābit ibn Qurra in Rashed [26] p. 337ff, and Rashed's analysis on page 159. Ṭābit computes an n -term sum by writing the terms to be summed, then below them $n - 1$ terms to be summed, and so on down to a single term. This produces a two-dimensional array, and Ṭābit computes the sum of the top line from properties of the whole array. For his exposition he takes $n = 4$ as a typical case. We saw that Ibn Sīnā takes cases of length 2 or 3, but here the parallel may break down, because we found that

these cases play a special role in the calculation.

Appendices

A Translation of *Qiyās* 9.6

IX.6 The analysis of syllogisms, with a mention of dos and don'ts that can be relied on and used in that [analysis].

Prior Analytics i.32 46b38

[9.6.1] Sometimes a person is addressed with a well-crafted and definitive syllogism, or he finds such a syllogism written in a book. But then [sometimes] the syllogism is not simple but compound; or it appears not as a connected whole but as scattered pieces. And sometimes moreover the pieces are jumbled out of their natural order, or a part of the syllogism is hidden, or something superfluous is added. [Even] when it is simple, sometimes it is jumbled out of its natural order, or missing a piece, or with a piece added. You already know how this happens. If we don't have rules to guide us, on how to seek with due deliberation the syllogism that proves a given goal, [and to confirm] the soundness of the connection between a given syllogism [and its goal], so that we can analyse the syllogism into a group of premises, put them in the natural order, strip off defects and add any part that is missing, reducing the syllogism to the syllogistic figure that produces it — [if we don't have rules for all this,] then the new information that the syllogism provides will escape us. If the syllogism is sound then [so is] what it entails. If it's faulty, one should locate the fault either in its premises or in its construction. 460.5 460.10

[9.6.2] So we need to have rules in the form of dos and don'ts, to be used in the analysis of a syllogism. The rules should apply, not on the basis that the syllogism is demonstrative or dialectical or some other kind, but on the basis that it is an absolute syllogism. Then when you are given [the syllogism], you reach what the analysis leads you to, and it agrees with your starting point when you followed the route of synthesis. Thus you find the truth agreeing with itself, however you come to it, and standing as witness to its essence. For the truth, insofar as it is what is the case, stands witness to its essence insofar as [its essence] is how the truth is conceptualised. Likewise insofar as [the essence of truth] is the starting point of [the 460.15 461.1

truth], [the truth] witnesses to its essence insofar as [the truth] is where [the essence] leads us to; and insofar as [the essence of truth] is where [the truth] leads us to, [the truth] stands as witness to its essence insofar as [the truth] is the starting point of [its essence].

[9.6.3] So when you have found a syllogism, you start by looking for its two premises. You do this before looking for the terms, because gathering up fewer things is easier [than gathering up many]. Also when you start with the terms, it can be that there are more than two ways of combining them into two premises, so that the cases you would need to consider would ramify. The reason for that is that by locating the terms you don't thereby locate the premises as things composed [from the terms]. You would have to examine the case of each term, and then examine four possible ways of combining [pairs of terms]. So you would have to consider five items: first you would consider the terms [themselves], and then you would consider the four cases which arise from the ways of composing the premises from two terms. But if you locate the two premises, it's enough for you to consider one more thing, namely to list the terms. Thus when you have found two premises, locating the syllogism and how it behaves will be easy for you. 461.4,5 461.10

[9.6.4] Then the first step is to investigate whether each of the premises shares one of its terms with the goal but is distinguished from the goal by another [term]. Suppose [it does, and] one of the two premises shares both its terms with one part of the second premise, while another part of the second premise — not the whole of it — shares both the terms of the goal. Then the syllogism is duplicative, and the premise which has one part overlapping the goal and another part overlapping the other premise is a propositional compound, while the other premise is a duplication. So look carefully at [the sentence] which has a part overlapping the goal in two terms: is it meet-like or difference-like? If it is meet-like then find out whether its overlap [with the goal] is its first or second clause, and find out whether that other [sentence] is the same [as this part of the premise], or is its contradictory. If the premise is difference-like, then find out whether the overlapping [clauses] are the same or contradictories. Do the same with the other [premise], which is the duplicating one. In this way your syllogism is analysed into the propositional moods. 461.12 462.1 462.5

[9.6.5] If this is not the case, and for every [sentence] of the syllogism the goal (which is proved through [the syllogism]) overlaps it in just one 462.5

term, then you know that the syllogism is recombinant. If you have found that each of the premises overlaps the conclusion, then look for the middle term, so that you find the figure. Then connect the terms to the conclusion, so as to find the major and minor [premises] and the other things that you should be looking for. If you can't find a middle term, then the syllogism is not simple; instead you have a compound syllogism with at least four terms. 462.10

[9.6.6] [First case: two given premises, each sharing one term with the goal]

[Problem 1.] Suppose the goal is universally quantified affirmative, namely 'Every C is an A ', and suppose that the found premises are 'Every C is a B ' and 'Every D is an A '. Then if it's clear that 'Every B is a D ', your syllogism is in good order; otherwise it needs a middle. 462.10

[Problem 2.] Suppose the goal is universally quantified negative, [namely 'No C is an A '], and suppose the found [premises] are 'Every C is a B ' and 'No D is an A '. Then consider whether 'Every B is a D '. If so, then a syllogism can be composed. If not, then it needs a middle. 462.12

[Problem 3.] Suppose the found premises are 'No C is a B ' and 'Every A is a D '. Then it will be no help to you in this case to find 'Every B is a D ', so that the negative [premise] becomes the minor [premise of a syllogism] in the first [figure] and the remaining two premises are affirmative. So consider whether it's true for you that 'Every D is a B '. If it is, then you say 'Every D is a B ' and 'No C is a B ', which entails 'No C is a D '. Then you add to it that 'Every A is a D ', so that it entails 'No C is an A '. 462.15 463.1

[Problem 4.] Suppose the found [premises] are 'No C is a B ' and 'Every D is an A '. Then it can't be used. 463.2

[Problem 5.] Suppose the goal is 'Some C is an A ', and you have found [the premises] 'Some C is a D ' and 'Every B is an A '. Then if 'Every D is a B ' is attached, you have found [the syllogism]. 463.2

[Problem 6.] If the found [premises] are 'Every D is a C ' and 'Every B is an A ', then if 'Every D is a B ' is attached, you have found [the syllogism]. 463.4

[Problem 7.] If the *ḥāṣil* [premises] are 'Every C is a D ' and 'Some B is an A ', then if 'Every D is a B ' or 'Some D is a B ' is attached, it can't be used. If 'Every C is a B ' or 'Some C is a B ' is attached, it can't be used. Likewise if 'Some B is a C ', or 'Some B is a D ' is attached, it can't be used. And likewise if 'Every B is a D ' is attached, it can't be used. And if 'Every B is a C ' is attached, it doesn't entail to ['Some] C is an A '. 463.5

[Problem 8.] If the found *ḥāṣil* [premises] are 'Some D is a C ' and 'Every B is an A ', and 'Every D is a B ' is attached, then this makes the syllogism 463.9

ḥāṣil.

[Problem 9.] If the *ḥāṣil* [premises] are ‘Every *D* is a *C*’ and ‘Every *B* is an *A*’, and ‘Every (or some) *D* is a *B*’ is attached, then this makes the syllogism *ḥāṣil*. 463.10

[Problem 10.] If the *ḥāṣil* [premises] are ‘Every *D* is a *C*’ and ‘Some *B* is an *A*’, it can’t be used. 463.11

[Problem 11.] If the *ḥāṣil* [premises] are ‘Some *D* is a *C*’ and ‘Every *A* is a *B*’, it can’t be used. 463.12

So consider the remaining cases [with existentially quantified affirmative goal] in the same way.

[Problem 12.] Suppose that the goal is existentially quantified negative: ‘Not every *C* is an *A*’, and that you have found [the premises] ‘Some *C* is a *B*’ and ‘No *D* is an *A*’. Then if [an appropriate sentence with terms] *B*, *D* is attached, then you can use it — for example ‘Every *B* is a *D*’. 463.13

[Problem 13.] If you have [the premises] ‘No *C* is a *B*’ and ‘Some *D* is an *A*’, it can’t be used. 463.15

[Problem 14.] Likewise if you have [the premises] ‘Every *C* is a *B*’ and ‘Not some *D* is an *A*’, [the syllogism can’t be used]. 463.16

[Problem 15.] If you have [the premises] ‘Not every *C* is a *B*’ and ‘Every *D* is an *A*’, then it can’t be used. 464.1

[Problem 16.] If you have [the premises] ‘Some *B* is a *C*’ and ‘No *D* is an *A*’, and ‘Every *B* is a *D*’ is attached, you can use it. 464.1

[Problem 17.] If [the premises] are ‘No *B* is a *C*’ and ‘Some *D* is an *A*’, it can’t be used. 464.3

[Problem 18.] If [the premises] are ‘Every *B* is a *C*’ and ‘Every *D* is an *A*’, it can’t be used. 464.3

[Problem 19.] If you have [the premises] ‘Not every *B* is a *C*’ and ‘Every *D* is an *A*’, it can’t be used. 464.4

[Problem 20.] If you have [the premises] ‘Some *D* is a *C*’ and ‘No *A* is a *B*’, and ‘Every *D* is a *B*’ is attached, then you can use it. 464.5

[Problem 21.] If you have [the premises] ‘No *C* is a *B*’, and ‘Some *A* is a *D*’, it can’t be used. 464.6

[Problem 22.] If the *ḥāṣil* [premises] are ‘Every *C* is a *B*’, and ‘Not some *A* is a *D*’, it can’t be used. 464.7

[Problem 23.] If the *ḥāṣil* [premises] are ‘Not every *B* is a *C*’, and ‘Every *A* is a *D*’, it can’t be used. 464.8

[Problem 24.] If you have: ‘Some *C* is a *B*’ and ‘No *A* is a *D*’, and ‘Every *B* is a *D*’ is attached, you can use it. 464.8

[Problem 25.] If you have [the premises] ‘No *B* is a *C*’ and ‘Some *A* is a *D*’, then it can’t be used. 464.10

[Problem 26.] If you have [the premises] ‘Every B is a C ’ and ‘Not every A is a D ’, it can’t be used. 464.10

[9.6.7] Likewise in the other remaining cases. This is when the two premises each share a term with the goal. If the two [premises] share [a term] with each other, and they don’t share with the goal at all, then don’t bother to analyse it, because in this case the shortfall is too great. And likewise when only one of the two shares [a term] with the goal, and the other doesn’t share with the goal or with its companion, then [the argument] is not straightforward to analyse. In order to explain how to analyse it we would need to apply a lengthy principle that is not expressible in a rule that one can take on board briefly. Analysis of [such an argument] is possible, but the appropriate place for this is the appendices, which will also [extend] analysis to more than two premises. 464.12
464.15
465.1

[9.6.8] [Second case.] If you have found two premises that share [a term] with each other, and one of them shares [a term] with the goal, then this shared [term] is either the subject or the predicate of the goal. Suppose it is the subject. 465.5

[Problem 27.] First suppose the conclusion is universally quantified and affirmative, thus: ‘Every C is an A .’ Suppose the found [premises] are ‘Every C is a B ’ and ‘Every B is a D ’. Then if you have found [a premise] linking D to A , this makes [the syllogism] *ḥāṣil*. 465.5

[Problem 28.] Suppose the conclusion is universally quantified negative [thus: ‘No C is an A ’], and the found [premises] are: ‘Every C is a B ’ and ‘Every B is a D ’. Then if you have found [the premise] ‘No D is an A ’, this makes [the syllogism] *ḥāṣil*. 465.7

[Problem 29.] If you have found [the premises] ‘Every C is a B ’ and ‘No B is a D ’, and then you found [the attachment] ‘Every A is a D ’, this makes [the syllogism] *ḥāṣil* without needing a conversion. 465.8

[Problem 30.] If you have found [the premises] ‘No C is a B ’ and ‘Every B is a D ’, it can’t be used. 465.10

[Problem 31.] If you have found [the premises] ‘No C is a B ’ and ‘Every D is a B ’, and then you found the premise ‘Every A is a D ’, this makes [the syllogism] *ḥāṣil*. 465.10

[Problem 32.] Suppose the conclusion is existentially quantified affirmative [thus: ‘Some C is an A ’]. Suppose [the premises] ‘Some C is a B ’ and ‘Every B is a D ’ are already *ḥāṣil*, and ‘Every D is an A ’ is attached, then this makes [the syllogism] *ḥāṣil*. 465.12

[Problem 33.] Suppose [we have] ‘Every D is a B ’ and ‘Every B is a C ’. 465.13

Then if ‘Every D is an A ’ or ‘Some D is an A ’ is attached, this makes [the syllogism] *ḥāṣil*.

[Problem 34.] Suppose [the premises] are ‘Every C is a B ’ and ‘Some B is a D ’; then this [syllogism] can’t be used. 465.14

[Problem 35.] If the existentially quantified [goal] is negative [thus: ‘Some C is not an A ’], and you have found [the premises] ‘Some C is a D ’ and ‘Every D is a B ’, and ‘No B is an A ’ is attached, this makes [the syllogism] *ḥāṣil*. 465.15

[Problem 36.] If you have found [the premises] ‘Some C is a B ’ and ‘No B is a D ’, and ‘Every A is a D ’ is attached, this makes [the syllogism] *ḥāṣil*. 466.1

Work through the remaining cases of this kind for yourself, taking the compound [syllogisms] in turn.

[9.6.9] You should know that when we said: ‘This makes [the syllogism] *ḥāṣil*’, this meant *ḥāṣil* without having to alter [the syllogism] by making a conversion in the found [premises]. Also you should know that we are not putting ourselves to the trouble of telling you now what figure the *ḥāṣil* [syllogism] is [proved] in. If you don’t understand that, and didn’t memorise what was said [about it earlier], you won’t have been able to make any use of this [lesson]. 466.3
466.5

[9.6.10] [Third case: Two premises which share one term with each other, and one of them shares a term with the predicate of the goal.]

[Problem 37.] If the shared [term] is in the predicate of the goal, and the goal is universally quantified affirmative [thus: ‘Every C is an A ’]; and you have [the premises] ‘Every D is a B ’ and ‘Every B is an A ’, and ‘Every C is a D ’ is attached, this makes [the syllogism] *ḥāṣil*. 466.6

[Problem 38.] If the goal is universally quantified negative [thus: ‘No C is an A ’], and the found [premises] are ‘Every D is a B ’ and ‘No B is an A ’, and ‘Every C is a D ’ is attached, this makes [the syllogism] *ḥāṣil*. 466.7

[Problem 39.] If the found [premises] that you have are ‘No D is a B ’ and ‘Every A is a B ’, and ‘Every C is a D ’ is attached, this makes [the syllogism] *ḥāṣil*. 466.9

[Problem 40.] If you have [the premises] ‘Every D is a B ’ and ‘No A is a B ’, and ‘Every C is a D ’ is attached, this makes [the syllogism] *ḥāṣil*. 466.10

[Problem 41.] If the goal is existentially quantified affirmative [thus: ‘Some C is an A ’], and you have [the premises] ‘Some B is a D ’ and ‘Every D is an A ’, and ‘Every B is a C ’ is attached, you can use it. 466.11

[Problem 42.] If you have: ‘Some B is a D ’, and ‘Every A is a D ’, it can’t be used. 466.13

- [Problem 43.] If you have ‘Some D is a B ’ and ‘Every B is an A ’, and [the attached premise] is ‘Every D is a C ’, you can use it. 466.13
- [Problem 44.] If you have ‘Some D is a B ’ and ‘Some A is a D ’, it can’t be used, even with the order [of the terms in a premise] converted. 466.14
- [Problem 45.] If your goal is existentially quantified negative [thus: ‘Some C is not an A ’], and you have [the premises] ‘Some B is a D ’ and ‘No D is an A ’, and ‘Every B is a C ’ is attached, you can use it. 466.15
- [Problem 46.] Or you have ‘Every B is a D ’ and ‘Some D is not an A ’ — then you can’t use it. 467.1
- [Problem 47.] If you have [the premises] ‘Not every B is a D ’ and ‘Every D is an A ’, you can’t use it. 467.2
- [Problem 48.] If you have ‘No B is a D ’ and ‘Some D is an A ’, you can’t use it. 467.2
- [Problem 49.] If you have ‘Some D is a B ’ and ‘No A is a B ’, and ‘Every D is a C ’ is attached, you can use it. 467.3
- [Problem 50.] If you have ‘No D is a B ’ and ‘Every A is a B ’, and ‘Some C is a D ’ is attached, you can use it. 467.4
- [Problem 51.] If you have ‘Not every D is a B ’, and ‘Some A is a B ’, it can’t be used. 467.5
- Try out for yourself the compound [syllogisms] where the overlap is with the predicate of the goal, in the same relation as above. 467.7
- These, and similar [examples] that we handle by comparison with them, are instances of analysis where you have two premises. 467.9
- [9.6.11] [Fourth case: One premise, which shares a term with the goal.]
- [Problem 52.] In the case where you have a single premise, which overlaps the predicate of the conclusion, and the goal is universally quantified affirmative, namely ‘Every C is an A ’, and you have [the premise] ‘Every D is an A ’, then if ‘Every C is a D ’ is attached, this makes [the syllogism] *ḥāṣil*. 467.9
- [Problem 53.] If you have ‘Every A is a D ’, it can’t be used. 467.12
- [Problem 54.] If the goal is universally quantified negative [thus: ‘No C is an A ’], and you have [the premise] ‘No D is an A ’ or ‘No A is a D ’, and ‘Every C is a D ’ is attached, this makes [the syllogism] *ḥāṣil*. 467.12
- [Problem 55.] If you have [the premise] ‘Every D is an A ’, then [the syllogism] can’t be made *ḥāṣil*. 467.14
- [Problem 56.] Rather, if you have ‘Every A is a D ’, and it’s true that ‘No C is a D ’, this makes [the syllogism] *ḥāṣil*. 467.14
- [Problem 57.] If the goal is existentially quantified affirmative [thus: 467.15

'Some <i>C</i> is an <i>A</i> ', and you have [the premise] 'Some <i>D</i> is an <i>A</i> ', and 'Every <i>D</i> is a <i>C</i> ' is attached, you can use it.	
[Problem 58.] If you have [the premise] 'Every <i>D</i> is an <i>A</i> ', and 'Some <i>C</i> is a <i>D</i> ' is attached, you can use it.	467.16
[Problem 59.] If you have 'Some <i>A</i> is a <i>D</i> ', you can't use it at all, unless you convert [the premise].	467.17
[Problem 60.] If the goal is existentially quantified negative [thus: 'Some <i>C</i> is not an <i>A</i> '], and you have [the premise] 'Every <i>D</i> is an <i>A</i> ', you can't use it at all.	467.18
[Problem 61.] Rather, if [the premise] is 'No <i>D</i> is an <i>A</i> ', and 'Some <i>C</i> is a <i>D</i> ' is attached, you can use it.	467.19 468.1
[Problem 62.] Likewise if you have 'Some <i>D</i> is an <i>A</i> ', it can't be used.	468.1
[Problem 63.] If you have [the premise] 'Not every <i>D</i> is an <i>A</i> ', and 'Every <i>D</i> is a <i>C</i> ' is attached, you can use it.	468.2
[Problem 64.] If [the premise] is 'Not every <i>A</i> is a <i>D</i> ', it can't be used.	468.3
[9.6.12] When you put the steps in this order, as I have shown you, you will reach the [required] terms, figures and moods. And the terms that you encounter will be ones within the formats mentioned above as ones that can be used.	468.4
Apply exactly the same considerations to propositional compounds.	468.7

B Notes on the text translated

The text above is translated from the Arabic text in [15], which is a volume from the Cairo edition of the *Šifā'*, published under the overall editorship of Ibrahim Madkour.

Title

Ibn Sīnā writes '*intafa^c X bi-Y*' to express 'X can use Y'. The passive form, which occurs in the title, is '*untufi^c bi-Y*', meaning 'Y can be used'. I haven't found this meaning in the dictionaries, including Goichon [9]. But it's fairly common in Ibn Sīnā's logical writing. For example in *Burhān* [16] 13.14 one can't use (*lam yantafi^c bi-*) what a teacher says unless one thinks for oneself; 63.8 there are students who can use (*intafa^c bi-*) a compass but are still stupid; 141.13 in debate one can't use (*lā yantafi^c bi-*) a proof that requires very many middle terms; *Ibāra* [14] 2.12f sciences are developed so that later generations can use (*yantafi^c bi-*) them. Dozy [7] comes nearest with the meaning

‘trouver son compte à’, which Gabriel Sabbagh kindly tells me can be translated as ‘finds advantageous or useful’.

[9.3.1]

460.5f ‘not connected but separated’ (*gair maḥṣūl bal maḥṣūl*): See subsections 4.2 and 4.4 above for these notions.

[9.6.2]

460.17 ‘absolute syllogism’ (*qiyās muṭlaq*): ‘absolute’ (*muṭlaq*) means without any restriction or condition being imposed. The kind of restriction he has in mind here is to syllogisms that are appropriate for a particular purpose. For example demonstrative syllogisms are for demonstrating that something is true by deducing it from things that are self-evident or already demonstrated, so their premises must be necessary truths. Dialectical syllogisms must have premises that are true for the most part and generally accepted. (*Qiyās* [15] p. 4, *Iṣārāt* [20] Method 9.)

461.2 ‘standing as witness to its essence’ (*ṣāhid li-dātili*): This rhetorical flourish apparently comes from the translator into Arabic, Taḍārī; it is not in the Greek original. Ibn Sīnā seems to have replied with a similar blossom of rhetoric. But was Taḍārī quoting something?

[9.6.3]

461.8 ‘as thing composed’ : i.e. rather than as segments of text.

Prior Analytics i.32 47a11 claims that it’s easier to divide a thing into large parts than into small, but offers no argument in support of this. Ibn Sīnā may be right about which order is easier, but his reason doesn’t convince. If you first locate the premises as segments of the text, you don’t thereby locate them ‘as composed from the terms’ either. There may be many different ways of carving a subject-predicate form out of one and the same sentence. For example if the sentence in front of you is ‘This line and that line meet’, should you parse it as ‘(This line) (meets that line)’ or as ‘(These two lines) (meet)’? Or to take Ramsey’s

more philosophical example, should you parse ‘Socrates is wise’ as ‘(Socrates) (is wise)’ or as ‘(Wisdom) (is a characteristic of Socrates)’ ([25] p. 21)? The nub of the matter is that Ibn Sīnā in this section ignores the possibilities of local formalising; cf. [13].

[9.6.4]

461.13 ‘its terms’: Ibn Sīnā is discussing propositional syllogisms here, so for example the ‘terms’ of the proposition ‘if p then q ’ are p and q , both of which are sentences and not terms in the usual sense. See subsection 4.5 above.

[9.6.6]

Problem 3. The two terms that occur once only in the given goal and premises are B and D , so we are looking for a sentence ϕ with terms B and D . The goal is universally quantified, so all the premises are universally quantified, and in particular ϕ is universally quantified. The goal is negative, so there is exactly one negative premise; hence the remaining two premises including ϕ must be affirmative. Thus ϕ must be either ‘Every B is a D ’ or ‘Every D is a B ’. We try both in turn. If we combine ‘Every B is a D ’ with ‘No C is a B ’ as the premises of a simple syllogism, then since B is subject in one and predicate in the other, the syllogism is in first figure, and its minor premise is ‘No C is a B ’ since this is the one with the middle term D as its predicate. But the only mood in first figure with two universally quantified premises and one of them negative is the second mood (*Celarent* in the Latin nomenclature), whose minor premise is affirmative. So ‘Every B is a D ’ can’t be used, and we have to try ‘Every D is a B ’ instead. The result is the following connected compound syllogism, which meets the requirements:

$$(27) \quad \begin{array}{c} \text{No } C \text{ is a } B. \quad \text{Every } D \text{ is a } B. \\ \hline \text{No } C \text{ is a } D. \quad \text{Every } A \text{ is a } D. \\ \hline \text{No } C \text{ is an } A. \end{array}$$

In his discussion Ibn Sīnā seems to derive ‘No C is a D ’ from the premises ‘Every D is a B ’ and ‘No C is a B ’ in that order, breaking the premise order condition. Probably the reason is in the immediately preceding text: ‘Consider whether . . . ‘Every D is a B ’. If it is, then’ (etc.) The premise ‘Every D is a B ’ follows on naturally from this, and Ibn Sīnā need not be claiming that it serves as first premise. (The case is quite different from Problem 33, where the order is unexpected until we remember Ibn Sīnā’s conventions.)

Problem 5. ‘found’: What is found? In Problems 8, 9 it’s explicitly the syllogism, and there are no examples where it’s explicitly the goal. So I infer the syllogism is meant here.

Problem 6. The problem is the same as Problem 9 below. The solutions are different; at Problem 9 Ibn Sīnā gives the weakest fill and one other, but here he gives only an unnecessarily strong fill. (See the end of subsection 4.4.) One might be tempted to change the text so as to remove the doublet. But there is another doublet: Problem 43 is Problem 41 with the letters B and D transposed. So I left the text alone.

Problem 7. Assuming the text is sound, here is a sequence of thoughts that it could represent.

First, the goal is affirmative, so there is no need to consider an added premise θ that is negative. Second, we saw in Problem 3 that θ should be taken with the first premise to yield an intermediate conclusion. The first premise is ‘Every C is a D ’, so either C or D is a term in θ . The other term can’t be A , since A already occurs twice; so it must be B . (This is clumsy: the same reasoning would eliminate C too, since it also occurs twice. Maybe Ibn Sīnā wanted his student to say ‘That’s clumsy’ and formulate the reason.) So first we try affirmative sentences with B as predicate. There are two with D as subject, and combining with ‘Every C is a D ’ they yield respectively ‘Every C is a B ’ and nothing at all. ‘Every C is a B ’ can’t combine with ‘Some B is an A ’, because it would give a first figure syllogism with existentially quantified major premise, which is impossible. There are two with C as subject, and they both give ‘Some B is a D ’ (or conversely). This can’t combine with ‘Some B is an A ’ because both are existentially quantified. Next we try the possibilities with B as subject. There are two existentially quantified, but again these

will yield an existentially quantified intermediate sentence that won't combine with 'Some B is a D '. There remain 'Every B is a D ' and 'Every B is a C '. The first yields nothing with 'Every C is a D '. The second yields 'Every B is a D ', which combines with 'Some B is an A ' to yield 'Some D is an A ', not the conclusion we want.

The last case, namely 'Every B is a C ', is important because it does in fact combine with the given premises to yield the required goal; but the resulting syllogism uses only the second premise. Since Ibn Sīnā doesn't count this as a solution of the problem, we have confirmation that the algorithm is not intended to eliminate unnecessary premises.

But there is also a problem about how to read the text at the end of the example. The Cairo edition has *lam yuntij 'ilā j*, which is meaningless. The sense has to be either that the inference using 'Some B is an A ' doesn't use the premise 'Every C is a D ', or that when used with all the other premises, it doesn't yield the required conclusion. The normal usage is that one entails *min* or *an* the premises and *alā* the conclusion; the preposition '*ilā*' 'to' could hardly be a variant of the first two, but it could be a variant of *alā* 'onto', though I've found no other examples. Hence we can guess that j should be $j a$, a shorthand for the goal (since our C corresponds to Arabic j). Therefore I propose to emend to *lam yuntij 'ilā j a*, and I have translated accordingly. See Problem 32 for the opposite error; there is very little difference between a handwritten Arabic a and a short scratch on the paper.

Problem 8. 'makes the syllogism *ḥāṣil*' (*qad ḥuṣṣila l-qiyās*): For this translation see the notes on Problem 32.

Problem 9. See on Problem 6.

Problem 14. If we add the premise 'Every B is a D ', then from 'Every C is a B ' we get 'Every C is a D '. To deduce 'Not every C is an A ' we need 'No D is an A '; 'Some D is not an A ' wouldn't be enough. So it looks as if Ibn Sīnā here understands 'Not some D is an A ' (*laisa ba^cdu d a*) as 'Some D is not an A '. At first sight this is at odds with his treatment of *laisa kullu*, which he always interprets as 'Not every'. There is a similar discrepancy at *Ibāra* [14] 67.10, where Ibn Sīnā says that *lā kull* and *lā ba^cd* are equivalent. The interactions of quantifiers and negation in Arabic are complicated; Jamal Ouhalla alerts me to the fact that focus can affect

scope. But as far as I can see, the relevant phrases in this Problem and at the ^c*Ibāra* reference are in topic position, though there are grounds for thinking that logic blinded Ibn Sīnā to questions of topic.

- Problem 22. Here ‘Not some A is a D ’ should be interpreted as ‘Some A is not a D ’. If it was read as ‘No A is a D ’ and we attached ‘Every B is a D ’, the first two premises would yield ‘Every C is a D ’, which combines with ‘No A is a D ’ to yield ‘No C is an A ’ by *Cesare*, and hence ‘Not every C is an A ’ by (8). Compare Problem 14.
- Problem 23. The Cairo text has ‘Every A is a D ’ (*kullu a d*) rather than ‘Every D is an A ’ (*kullu d a*). But with that reading Ibn Sīnā should say that attaching ‘Every D is a B ’ yields the required syllogism. I’ve adopted the easiest correction that doesn’t introduce a doublet.

[9.6.8]

465.2 ‘in the appendices’ (*bil-lawāḥiq*): In several places in the *Šifā* Ibn Sīnā refers to things that will appear in the appendices. But no work of this name or with exactly the required contents has been found. It has been suggested that Ibn Sīnā’s two other works *Ta^clīqāt* and *Mubāḥatāt* contain material that was intended for the appendices. (Gutas [12] pp. 141–144.) But the published versions of these two works contain only philosophical material, and nothing about proof search. More’s the pity, because Ibn Sīnā’s treatment of incomplete syllogisms with two or more gaps would have shown us more about how he handled problems of search. See subsection 6.2 for more on the historical context.

- Problem 27. Ibn Sīnā doesn’t say what premise linking D to A will work. There may be a subtle reason. This is the first example with two premises ϕ_1, ϕ_2 to the left of the gap, so the student has a choice between first combining ϕ_1 with ϕ_2 before combining the result with the test sentence; or first combining ϕ_2 with the test sentence and then bringing in ϕ_1 . The first route is clearly more sensible, because the result of combining ϕ_1 with ϕ_2 will be the same for each test sentence. Ibn Sīnā forces the student to see this, by putting pressure on the student to try several test sentences. But the effect is slightly spoiled by the fact that in this particular case the answer ‘Every D is an A ’ is obvious without any calculation.

- Problem 29. Paragraph [9.6.9] below suggests that Ibn Sīnā is talking about converting a premise. But why should anybody think of converting a premise in this example? A possible explanation lies in the fact that this is the first example in this block where a premise has its terms out of the obvious order. We might expect $(C, B)(B, D), (D, A)$, but instead the last premise gives (A, D) . Perhaps Ibn Sīnā had students who (apparently like Smith [3] Note to 42b5–26) assumed that switches like this don't occur. Ibn Sīnā had made the same point already at *Qiyās* 444.5, where it seems to have confused the copyists.
- Problem 32. The Arabic contains two occurrences of *qad ḥuṣṣil*, but they must mean different things. In general *qad* with the past tense is a perfective marker: it indicates that the present state is the outcome of a previous action described by the verb. But previous to what? At the first occurrence here the phrase must mean previous to the problem having been posed, hence 'already *ḥāṣil*'. But the second occurrence describes the outcome of the algorithm, so it can't mean that; it must mean that the application of the algorithm created the present situation, hence 'this makes it *ḥāṣil*'. Also the 'a' at the end of the sentence in line 465.13 should be deleted (as in one ms); cf. Problem 7.
- Problem 33. This is the one Problem where Ibn Sīnā gives the premises in an order that doesn't form a linkage where the goal subject points leftwards and the goal predicate points rightwards. The reason for this is explained in subsection 4.4 above.
- The solution 'Every *D* is an *A*' is redundant since it implies 'Some *D* is an *A*', which is already a solution (it's the weakest fill). Ibn Sīnā's procedure of trying all options is likely to throw up redundancies of this kind. But maybe he expects his brighter students to note the redundancy and formulate a policy.
- Problem 34. Delete the second occurrence of *wa-ba^cd b d*.
- 466.1 'taking the compound [syllogisms] in turn' (*bi-ḥasabi l-tarākīb*): *ḥasb* means calculation (as in the modern *ḥāsib* 'computer'). But there is probably no reference to computing or algorithms here. 'Calculations' in Ibn Sīnā's time were normally assumed to be numerical. So *bi-ḥasab* here probably has its usual meaning of 'according to'.

[9.6.10]

- Problem 37. Delete *mādā* at the beginning of line 466.6. Also the 'i' in the Cairo edition is a misprint for 'in.
- Problem 44. The student might worry that these two premises violate the fourth figure condition. Strictly this is not relevant, because the connected syllogism wouldn't combine these two premises in a simple syllogism; but it may explain why Ibn Sīnā remarks that no conversion is needed.
- Problem 48. In the Cairo text the first premise is 'No *B* is a *C*', violating the case assumption for [9.6.10]. Read 'No *B* is a *D*', following two manuscripts.
- Problem 51. The Cairo text has 'Every *A* is a *B*' for the second found premise. There must be a slip, because in that case we get a syllogism by attaching 'Every *D* is a *C*'. But on the Cairo reading this is also the only example in this block where the second found premise is the same as in the previous example. So I have replaced 'Every *A*' by 'Some *A*'.

[9.6.11]

- Problem 59. The only sentence that will complete the syllogism logically is 'Every *D* is a *C*'. The middle term is *D*, which is subject in 'Every *D* is a *C*' and predicate in 'Some *A* is a *D*', so the syllogism violates the fourth figure condition. Converting the premise 'Some *A* is a *D*' to 'Some *D* is an *A*' yields a third-figure syllogism in *Disamis*.
- Problem 61. We have to correct 'Some *C* is an *A*' to 'Some *C* is a *D*'.
- Problem 62. The Cairo text reads 'Likewise if [the premise] is 'No *A* is a *D*', and you have (*indak*) 'Some *D* is an *A*' or 'Some *A* is a *C*', it can't be used.' There are several problems with this. First, with the datum 'No *A* is a *D*' we get the goal by appending 'Some *C* is a *D*'; so the datum is presumably wrong. Second, this is the one problem where Ibn Sīnā seems to introduce the appended sentence with *indak*; in 28 other problems *indak* introduces the datum. Third, the sentence 'Some *A* is a *C*' is silly here, because it has the same terms as the goal. We can get a reasonable problem by deleting the first and third syllogistic sentences and the

text around them, as I have done in the translation. Then Ibn Sīnā is saying correctly that the goal can't be reached from the datum 'Some D is an A '.

[9.6.12]

468.7 'propositional connectives': An example of a simple recombinant propositional syllogism, using the propositional connective 'If ... then', is

(28) If p then q . If q then r . Therefore: If p then r .

where p , q and r are declarative sentences. Ibn Sīnā brings this to a form analogous to a predicative syllogism by the device of 'replacing "if" by "whenever"' (e.g. *Qiyās* 471.5), so that the sense becomes

Every occasion on which p is an occasion on which q .
(29) Every occasion on which q is an occasion on which r .
Therefore: Every occasion on which p is an occasion on which r .

This reduction, together with similar ones for some other propositional connectives, allows the proof search algorithm to be carried over routinely from predicative syllogisms to propositional ones.

C The ASM

Briefly, an ASM consists of a set of rules operated by modules; for example the module PROOFSEARCH below has four modules, namely DESCRIBE, SYNTHESISE, RAMIFY and SELECT. At each step in a computation, *all* the rules are applied once and simultaneously; if they clash, the machine stops. Rules normally begin with a condition, so they do nothing unless the condition is met. They can activate other rules by resetting parameters so that the conditions for the other rules are met. For example when the conditions for SYNTHESISE are met, the rules of SYNTHESISE have the effect of shortening the datum by 1 every time they operate. They continue to operate until there are no consecutive formulas in the datum with a term in common; at this point the condition for SYNTHESISE fails, but that for RAMIFY may be met, so that the rules of RAMIFY take over.

The notation $X := Y$ means that the value of the parameter X becomes Y . The notation X^* means the set of nonempty finite sequences of elements of X . I hope the rest is reasonably self-explanatory.

The logical part of this ASM was implemented in Perl 5 and run on all of Ibn Sīnā's 64 problems. There were discrepancies from Ibn Sīnā's solution (as reported in the Cairo text) at Problems 23, 33, 51, 61 and 62. These are all discussed in the notes to the relevant problems in section B above. Problem 33 is the only one of the 64 problems that lies outside the domain of the algorithm. The other four discrepancies are well within the range of transcription errors that one might expect from the state of the manuscript tradition. Some could possibly be misprints in the Cairo edition — I haven't consulted any manuscripts to check this.

Multi-agent ASM

Universe: *INTELLECT* (dynamic set of agents)

```
(ASM1) ACTIVEINTELLECT=  
  forall  $\iota \in INTELLECT$  if  $next(currpage).\iota = needsmiddle$  then  
    let  $k = |hasils(fill(currpage).\iota)|$   
    if  $k > 0$  then  
      let  $\iota_k := \iota$   
      if  $k > 1$  then  
        let  $\iota_1, \dots, \iota_{k-1} = new(INTELLECT)$   
        forall  $1 \leq i \leq k$   
          let  $p = new(PAGE).\iota_i$   
          datum( $p$ ). $\iota_i := insert(datum(currpage).\iota_i,$ 
```

$$\begin{aligned}
& i\text{-th}(\text{hasils}(\text{fill}(\text{currpage}).\iota_i), \text{gapsite}(\text{currpage}).\iota_i) \\
& \text{gapsite}(p).\iota_i := \text{needscalculating} \\
& \text{previous}(p).\iota_i := \text{currpage}.\iota_i \\
& \text{next}(p).\iota_i := 0 \\
& X(p).\iota_i := X(\text{currpage}).\iota_i \\
& \text{where } X = \text{edges}, \text{fill} \\
& \text{currpage}.\iota_i := p
\end{aligned}$$

where $\text{hasils}(\phi).\iota$ is the set of all sentences that are *hāṣil* for the intellect ι and have exactly one term in common with ϕ ; we assume this set is finite. In general the i -th sentence in the set will contain a term t that is not already in $TERM.\iota_i$, so the Active Intellect will need to add t to $TERM.\iota_i$; the term t is the imprinted form that we met in (26).

Signature of the agent ASMs

Universes: $TERM, PAGE$

Globals:

$$\begin{aligned}
& \text{goal} \in SENTENCE \\
& \quad (\text{input from the problem}) \\
& \text{currpage} \in \mathbb{N} \\
& \quad (\text{initially } 0) \\
& \text{report} \in \{\text{ignorance}, \text{logical failure}, \text{logical success}, \text{tahsil success}\} \\
& \quad (\text{initially} = \text{ignorance})
\end{aligned}$$

Page properties:

$$\begin{aligned}
& \text{datum} : PAGE \rightarrow SENTENCE^* \\
& \quad (\text{initially input from the problem}) \\
& \text{gapsite} : PAGE \rightarrow \mathbb{N} \cup \{\text{needscalculating}\} \\
& \quad (\text{initially} = \text{needscalculating}) \\
& \text{edges} : PAGE \rightarrow TERM^2 \\
& \text{fill} : PAGE \rightarrow SENTENCE \\
& \text{previous} : PAGE \rightarrow PAGE \\
& \text{next} : PAGE \rightarrow PAGE \cup \{\text{needsmiddle}\} \\
& \quad (\text{initially } 0)
\end{aligned}$$

Agent modules

(ASM2) PROOFSEARCH = {DESCRIBE, SYNTHESISE, RAMIFY, SELECT}

(ASM3) DESCRIBE =

```
if  $gapsite(currpage) = needscalculating$  then  
  let  $p = new(PAGE)$   
  take  $datum(currpage)$ ,  $goal$  and identify the gap site, the left  
  edge and the right edge. (If no gap then the gap site is 0.)  
   $gapsite(p) :=$  calculated gap site  
   $edges(p) :=$  (left edge, right edge)  
   $previous(p) := currpage$   
   $X(p) := X(currpage)$   
  where  $X = datum, fill, next$   
   $currpage := p$ 
```

(ASM4) SYNTHESISE =

```
if  $gapsite(currpage) \geq 0$  and  $next(currpage) \geq 0$  and  
  ( $length(datum(currpage)) > 2$  or  
  ( $length(datum(currpage)) = 2$  and  $gapsite(currpage) \neq 1$ )) then  
  let  $k = \begin{cases} 1 & \text{if } gapsite(currpage) \neq 1, \\ 2 & \text{otherwise} \end{cases}$   
  let  $\ell = length(datum(currpage))$   
  let  $\phi = consequence(k\text{-th}(datum(currpage)),$   
     $(k+1)\text{-th}(datum(currpage)))$   
  if  $\phi \neq sterile$  then  
    let  $\alpha = replacepair(datum(currpage), \phi, k)$   
    let  $p = new(PAGE)$   
     $datum(p) := \alpha$   
     $previous(p) := currpage$   
    if  $gapsite(currpage) > 1$  then  
       $gapsite(p) := gapsite(currpage) - 1$   
    else  
       $gapsite(p) := gapsite(currpage)$   
     $X(p) := X(currpage)$   
    where  $X = edges, fill, next$   
     $currpage := p$   
  else  
    if  $next(currpage) > 0$  then  
       $currpage := next(currpage)$ 
```


else
if *report = ignorance* **then**
report := logical failure

(ASM5) RAMIFY=

if *gapsite(currpage) > 0 and next(currpage) ≥ 0 and*
(length(datum(currpage)) ≤ 1 or
(length(datum(currpage)) = 2 and gapsite(currpage) = 1)) then
if *length(datum(currpage)) ≥ 1 then*
let $p_1, \dots, p_8 = \text{new}(PAGE)$
forall $1 \leq i \leq 8$
let $\phi = \text{listsentences}(\text{1-th}(\text{edges}(\text{currpage})),$
 $\text{2-th}(\text{edges}(\text{currpage})), i)$
 $\text{datum}(p_i) := \text{insert}(\text{datum}(\text{currpage}), \phi, \text{gapsite}(\text{currpage}))$
 $\text{fill}(p_i) := \phi$
 $\text{gapsite}(p_i) := 0$
 $\text{edges}(p_i) := \text{edges}(\text{currpage})$
 $\text{previous}(p_i) := \text{currpage}$
forall $1 \leq j \leq 7$
 $\text{next}(p_j) := p_{j+1}$
 $\text{next}(p_8) := 0$
 $\text{currpage} := p_1$
else
let $p = \text{new}(PAGE)$
 $\text{datum}(p) := \text{insert}(\text{datum}(\text{currpage}), \text{goal}, 1)$
 $\text{gapsite}(p) := 0$
 $\text{edges}(p) := \text{edges}(\text{currpage})$
 $\text{fill}(p) := \text{goal}$
 $\text{previous}(p) := \text{currpage}$
 $\text{next}(p) := 0$

(ASM6) SELECT=

if *gapsite(currpage) = 0 and length(datum(currpage)) = 1 then*
if *1-th(datum(currpage)) = goal then*
if *hasil(fill) = true then*
report := tahsilsuccess
else
let $k = \text{least } k \geq 1 \text{ such that}$
 $\text{gapsite}(\text{previous}^k(\text{currpage})) > 0$

```

let  $p := new(PAGE)$ 
 $X(p) := X(previous^k(currpage))$ 
  where  $X = datum, gapsite, edges$ 
 $fill(p) := fill(currpage)$ 
 $next(p) := needsmiddle$ 
 $previous(p) := currpage$ 
 $report := logicalsuccess$ 
 $currpage := p$ 
else
  if  $next(currpage) > 0$  then
     $currpage := next(currpage)$ 
  else
    if  $report = ignorance$  then
       $report := logicalfailure$ 

```

Basic functions

(Def1) $SENTENCE \subseteq TERM^2 \times \{0, 1\}^2$
 $SENTENCE(s, t, i, j) \Leftrightarrow s \neq t$

(Def2) $listsentences : TERM^2 \times \{1, \dots, 8\} \rightarrow SENTENCE$
 $listsentences(s, t, 1) = (s, t, 0, 0)$
 $listsentences(s, t, 2) = (s, t, 0, 0)$
 $listsentences(s, t, 3) = (s, t, 0, 1)$
 $listsentences(s, t, 4) = (s, t, 0, 1)$
 $listsentences(s, t, 5) = (s, t, 1, 0)$
 $listsentences(s, t, 6) = (s, t, 1, 1)$
 $listsentences(s, t, 7) = (s, t, 1, 1)$
 $listsentences(s, t, 8) = (s, t, 1, 1)$

(Def3) $consequence : SENTENCE^2 \rightarrow SENTENCE \cup \{sterile\}$
 $consequence(\phi, \psi) =$
 $\begin{cases} \text{strongest consequence of } [\phi, \psi] & \text{if } [\phi, \psi] \text{ is not sterile,} \\ sterile & \text{otherwise.} \end{cases}$

(Def4) $replacepair : SENTENCE^* \times SENTENCE \times \mathbb{N} \rightarrow SENTENCE$
 $replacepair([\phi_1, \dots, \phi_n], \psi, i) = [\phi_1, \dots, \phi_{i-1}, \psi, \phi_{i+2}, \dots, \phi_n]$.

(Def5) $insert : SENTENCE^* \times SENTENCE \times \mathbb{N} \rightarrow SENTENCE$

$insert([\phi_1, \dots, \phi_n], \psi, i) = [\phi_1, \dots, \phi_i, \psi, \phi_{i+1}, \dots, \phi_n]$.

(Def6) $hasil : SENTENCE \rightarrow \{true, false\}$

a user-defined basic function

References

- [1] Alcinoos, *Enseignement des Doctrines de Platon*, ed. John Whittaker, Budé, Paris 1990.
- [2] Alexander of Aphrodisias, *On Aristotle Prior Analytics 1.32–46*, translated by Ian Mueller, Duckworth, London 2006.
- [3] Aristotle, *Prior Analytics*, translated and edited by Robin Smith, Hackett, Indianapolis Indiana 1989.
- [4] Norman L. Biggs, E. Keith Lloyd and Robin J. Wilson, *Graph Theory 1736–1936*, Clarendon Press, Oxford 1976.
- [5] Egon Börger and Dean Rosenzweig, ‘A mathematical definition of full Prolog’, *Science of Computer Programming* 24 (1995) 249–286.
- [6] Egon Börger and Robert Stärk, *Abstract State Machines*, Springer, Berlin 2003.
- [7] R. P. A. Dozy, *Supplément aux Dictionnaires Arabes*, Librairie du Liban, Beirut 1968.
- [8] Sten Ebbesen, *Commentators and Commentaries on Aristotle’s Sophistici Elenchi, Vol. 1, The Greek Tradition*, Brill, Leiden 1981.
- [9] A.-M. Goichon, *Lexique de la Langue Philosophique d’Ibn Sīnā*, Desclée de Brouwer, Paris 1938.
- [10] Yuri Gurevich, ‘Evolving algebras. A tutorial introduction’, *Bulletin of european Association for Theoretical Computer Science* 43 (1991) 264–284.
- [11] Yuri Gurevich, Margus Veanes and Charles Wallace, ‘Can abstract state machines be useful in language theory?’, *Theoretical Computer Science* 376 (2007) 17–29.

- [12] Dimitri Gutas, *Avicenna and the Aristotelian Tradition: Introduction to Reading Avicenna's Philosophical Works*, Brill, Leiden 1988.
- [13] Wilfrid Hodges, 'Traditional logic, modern logic and natural language', *Journal of Philosophical Logic*, volume in honour of Johan van Benthem, ed. Hans von Ditmarsch and Larry Moss (to appear).
- [14] Ibn Sīnā, *Al-^cIbāra*, ed. Ibrahim Madkour et al., Dār al-Kātib al-^cArabī lil-Ṭabā^c wal-Našr, Cairo 1970.
- [15] Ibn Sīnā, *Al-Qiyās*, ed. Ibrahim Madkour et al., Našr Wizāra al-Ṭaqāfa wal-'Iršād al-Qūmī 1964 (referred to above as the Cairo edition).
- [16] Ibn Sīnā, *Al-Burhān*, ed. ^cAbdurrahmān Badawī, Dār al-Nahḍa al-^cArabīyya, Cairo 1966.
- [17] Ibn Sīnā, *Al-Sufista*, ed. Ibrahim Madkour et al., Našr Wizāra al-Ṭaqāfa wal-Ta^clīm, Cairo 1956.
- [18] Ibn Sīnā, *Al-Najāt*, Jamī^c al-Ḥuqūq, Beirut 1992.
- [19] Ibn Sīnā, *Mantiq al-Mašriqiyyīn*, Al-Maktaba al-Salafiyya, Cairo 1910.
- [20] Ibn Sīnā, *Al-Išārāt wal-Tanbiyyāt*, ed. Mojtaba Zāre^cī, Būstān-e Ketāb-e Qom, Qom, Iran 2000. (The logical part is translated: Shams C. Inati, *Ibn Sīnā, Remarks and Admonitions, Part One: Logic*, Pontifical Institute of Mediaeval Studies, Toronto 1984.)
- [21] F. Jabre, *Al-Naṣṣ al-Kāmil li-Mantiq Aristū* vol. 1, Dār al-Fikr al-Libnānī, Beirut 1999.
- [22] Victor J. Katz, 'Combinatorics and induction in medieval Hebrew and Islamic mathematics', in *Vita Mathematica: Historical Research and Integration with Teaching*, ed. Ronald Calinger, Mathematical Association of America 1996, pp. 99–106.
- [23] W. Kutsch, 'Muḥaṣṣal – Ġayr Muḥaṣṣal', *Mélanges de l'Université Saint Joseph* 27 (1947–8) 169–176.
- [24] John Philoponus, *In Aristotelis Analytica Priora Commentaria*, ed. M. Wallies, Reimer, Berlin 1905.
- [25] F. P. Ramsey, *Foundations: Essays in Philosophy, Logic, Mathematics and Economics*, ed. D. H. Mellor, Routledge & Kegan Paul, London 1978.

- [26] Roshdi Rashed, *Les Mathématiques Infinitésimales du IXe au XIe Siècle, Vol. I Fondateurs et Commentateurs*, Al-Furqān, London 1996.
- [27] Roshdi Rashed, *Al-Khwārizmī, Le Commencement de l'Algèbre*, Blanchard, Paris 2007.
- [28] W. D. Ross, *Aristotle's Prior and Posterior Analytics*, Clarendon Press, Oxford 1949.
- [29] Nabil Shehaby, *The Propositional Logic of Avicenna*, Reidel, Dordrecht 1973.
- [30] Tony Street, 'An outline of Avicenna's syllogistic', *Archiv für Geschichte der Philosophie* 84 (2) (2002) 129–160.
- [31] Paul Thom, *The Syllogism*, Philosophia Verlag, Munich 1981.
- [32] Kees Versteegh, *Landmarks in Linguistic Thought III: The Arabic Linguistic Tradition*, Routledge, London 1997.
- [33] Ernst Zermelo, 'Untersuchungen über die Grundlagen der Mengenlehre I', *Mathematische Annalen* 65 (1908) 261–281.