Chapter 5
Transition Networks

Transition networks (*TN*) are made up of a set of finite automata and represented within a graph system. The edges indicate transitions and the nodes the states of the single automata. Each automaton stands for a non-terminal symbol and is represented by its own network. The edges of each single network are denoted by non-terminal or terminal symbols and thus refer to other networks or final states. If the structure of a transition network also allows for recursive processes, for example, in the substitution of an object by another object belonging to a higher hierarchy level (e.g. a verb becomes a verbal phrase), this type of network is known as a *recursive transition network*. A path traversing the transition network starts at a first network and, beginning at the starting node, passes along the single edges. When it encounters a non-terminal symbol, the system branches like a sub-program to the corresponding network until finally all non-terminal symbols have been substituted. If different substitution possibilities are available, several paths between starting state and final state of the respective finite automaton exist. Figure 5.1 shows a transition network for expressions in natural language which may generate expressions such as "conductor likes singer," "a singer hates the conductor," "a singer likes a conductor hates the singer".

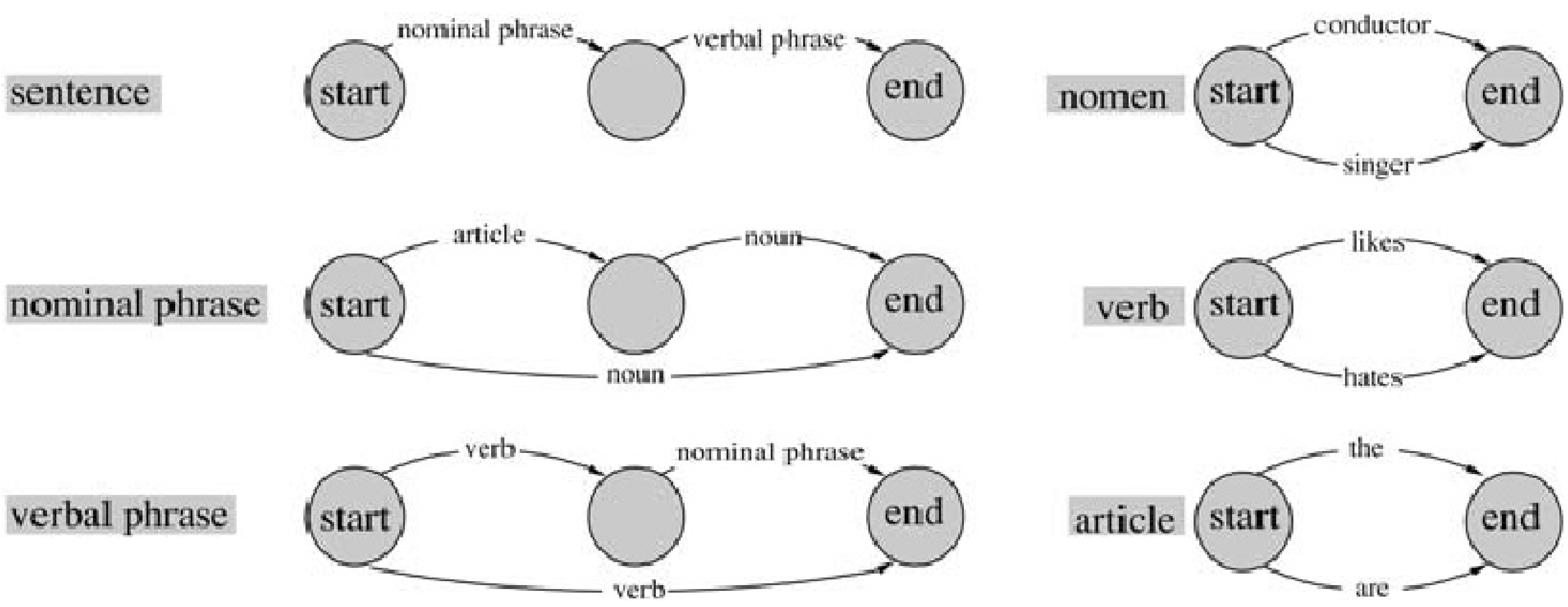

Fig. 5.1 A transition network for natural language expressions.

In an *augmented transition network* (*ATN*), the TN is extended in a way that allows specific instructions, conditional jumps or also whole sub-programs to be assigned to the edges. Augmented transition networks which were developed in the 1960s [11], are equivalent to type-0 grammars in terms of their generative capacity. Figure 5.2 illustrates an example of a simple ATN for the generation of melodic phrases. As an additional condition, the command "jump" is introduced here, inducing the omission of the current node. The upper and lower half of the graph enable generations in differing meters; a possible bass accompaniment is indicated below by means of some segments.

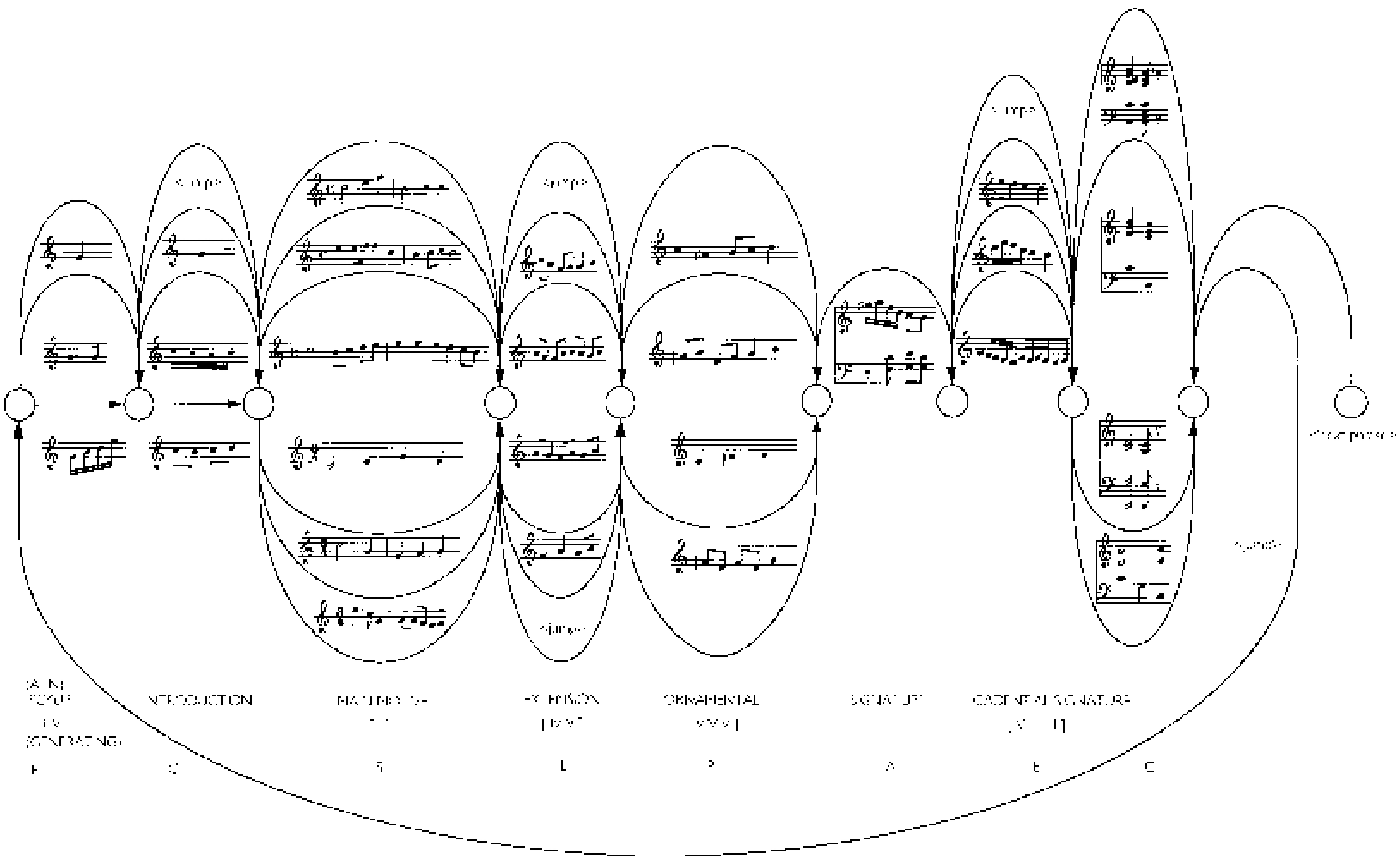

Fig. 5.2 Simple ATN for the production of musical segments [2, p. 64–65]. Reproduced with kind permission by A-R editions.

5.1 Experiments in Musical Intelligence

David Cope's Experiments in Musical Intelligence (EMI) is a well-known system of algorithmic composition which generates compositions conforming to a given musical style. Since its creation in 1981, Cope has continuously advanced EMI and described it in great detail in a number of publications.[1] EMI joins a series of different approaches to musical structure genesis and is also often mentioned in the context of artificial intelligence – Cope himself presents his system amongst oth-

[1] E.g. articles [1], [3]; book publications [2], [4], [5], [6].

ers in the framework of a "musical Turing test."[2] In view of the efficiency of EMI, Douglas Hofstadter revised some of his assumptions regarding the musical-creative potential of computer programs.[3] First experiments with a rule-based system for the generation of four-part movements Cope found unsatisfactory; as a consequence, he developed the approach of musical "recombinancy": In analogy to the historical model of the musical dice game, musical components are arranged to form a new composition, but with the essential difference being that EMI detects the components autonomously by means of the complex analysis of a corpus, transforms them partly and recombines them in an extensive process. For Cope, this principle also represents the implementation of a personal musical credo: "This program thus parallels what I believe takes place at some level in composers minds, whether consciously or subconsciously. The genius of great composers, I believe, lies not in inventing previously unimagined music but in their ability to effectively reorder and refine what already exists." [4, p. 93ff]. Since Cope implements the complex strategies of recombination within an augmented transition network, descriptions of basic functions of EMI are covered in this chapter.

Fig. 5.3 David Cope and Douglas Hofstadter. With kind permission of David Cope and Douglas Hofstadter.

In initial experiments, Cope divides up Bach chorales into single harmonic segments and recombines them by considering the correct voice leading, using only the transitions of harmonic segments which also occur in the original chorales. To provide sufficient material for new generations, the chorales of the corpus are transposed to a single key before analysis; the key for new compositions is chosen with regard to occurring voice ranges and the like. Although this simple principle of combination creates correct chorale progressions in either case, an acceptable structuring of a whole composition cannot be achieved. Due to this, additional strategies are applied that treat the musical material under numerous aspects for analysis and generation. In order to obtain a universally acceptable structure, a chorale of the

[2] See chapter 10; for Cope's test, cf. section "The Game" in [6, p. 33ff].

[3] Hofstadter refers here to his own prognoses in his book "Gödel, Escher, Bach: an Eternal Golden Braid" [8], cf. Hofstadter's essay "Staring Emmy Straight in the Eye ...," in [6, p. 33ff].

corpus may serve as a model for the sequence of phrases, cadence progressions, and the like. This model of chorale, however, only represents a meta-structure which is assigned with concrete musical segments from the EMI database. This database contains the complete material of the corpus, divided up into musical segments of different meaning. For the extension of the database and consequently also to increase the generalization power of EMI, coherent musical variations may be generated from segments of the corpus.[4] In order to be able to analyze, represent and process musical information in terms of different aspects, the following components and strategies are applied: SPEAC, an analysis model that analyzes the formal meaning of musical segments of different length on different hierarchical levels and makes them accessible for resynthesis; recognition and indication of characteristic and form-determining movements for their application in original or modified form in the new generations; implementation of the recombination strategies within an augmented transition network.

The musical units are recombined to form compositions in the relevant style in accordance with syntactic and "semantic" criteria. Here, the syntax describes allowed combinations of the terminals, whereas "semantics" guarantee that these also fulfil reasonable formal functions on their positions. Syntactic correctness in the combination of musical units can be seen through melodic components, for example in the transitions between the single parts. If, for example, a particular phrase in the corpus passes to another phrase through a major second movement, then this voice leading will also be retained in the recombination. This means that, in this case, only melodic segments are used as successors that may be reached from the last note of the previous phrase through a major second step. The syntactic correctness of a structure generated this way may be compared with the well-formedness of an expression generated by a generative grammar. Irrespective of that, the problem of "semantically" coherent meaning must be solved. If a generated expression is musically meaningful in the sense of musical semantics, it is defined in Cope's work by a sequence of musical components that due to their respective positions fulfil coherent forming and structuring functions. In functional harmony, for example, cadences that are characterized by dominant-tonic progressions would follow the "semantic" scheme of tension and relaxation. The imprecise terms "tension" and "relaxation" are used here according to Cope's classification system, which uses a general terminology to allow description of musical aspects on different formal levels. The semantic classification of the material is carried out by a system called SPEAC which is an acronym of the terms "statement," "preparation," "extension," "antecedents" and "consequent." These elements are used to indicate the musical units and denote relations that these may have to each other. "Statements" represent units that do not exist as the result of a particular process, but are included in a context in the course of the processing. "Preparations" are introductory gestures that stand ahead of other components and modify their meaning. "Extensions" follow other units (other than "preparations" or "extensions") and extend the preceding musical material. "Antecedents" prepare a concrete musical situation and demand

[4] Such as e.g. diatonic transposition which may represent a musical segment in the framework of different interval constellations; cf. [6, p. 102ff].

"consequents" as a solution. Figure 5.4 shows the different categories of SPEAC by means of a harmonic progression.

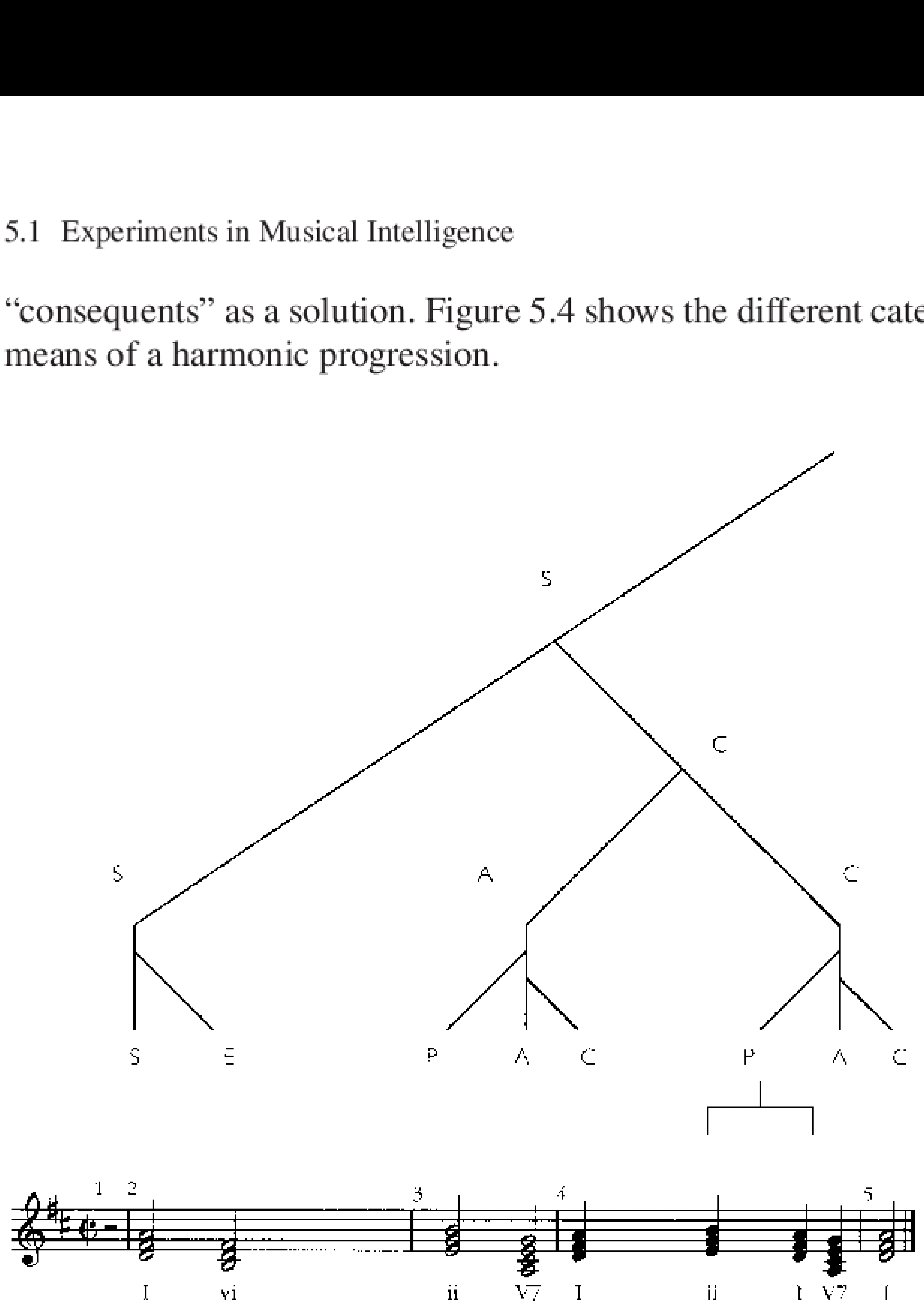

Fig. 5.4 Example of a harmonic parsing by SPEAC [2, p. 34]. Reproduced with kind permission by A-R editions.

The tonic parallel (h minor) is, for example, considered as the extension of the tonic due to its third relationship with D major. In this structure, the dominant seventh chord A7 in measure three is the "preparation" and the tonic "consequent." Here, the abstractions of SPEAC are assigned to the musical units on the basis of the intervallic constellation, the metric position and the duration of the single events. The order of the components of SPEAC is restricted by the following rules that indicate the possible successors of a component:

S → P, E, A
A → E, C
P → S, A, C
C → S, P, E, A
E → S, P, A, C

The SPEAC system is inspired by the analysis methods developed by Heinrich Schenker (see chapter 4) and allows for the interpretation of musical segments on different hierarchical levels. The dominant seventh chord in figure 5.4, for example, represents in the context of the two adjacent chords the function "antecedent" after "preparation" and ahead of "consequent." On a higher hierarchical level, these three chords represent an "antecedent" after a "statement" (measure 2) and before the final cadence which acts as a "consequent." This segmentation continuously involves increasing context dependencies in the examination, until finally a complete composition of the corpus is indicated as a first "statement," similar to the starting symbol "S" in a generative grammar. If during recombination a musical terminal[5] should be placed at a particular position, it is selected from the corpus according to syntactic conditions as well as the semantic context. In case, for example, the musical terminal should fulfil the function "PASC"[6] at this specific position, it is precisely these terminals chosen from the corpus that are designated with the classification "PASC" by EMI. If this selection turns out to be unsatisfactory, terminals with same properties but on a lower formal hierarchical level are searched, namely musical segments that fulfil the function "PAS," etc. If the formal context of the musical units in the corpus corresponds to that in the recombination, it must also be considered that, in order to achieve innovative solutions, the order of the terminals selected for new generations does not correspond to longer segments of the corpus.

To increase style conformity, EMI, during the analysis and recombination, also processes musical material which is of specific structuring meaning. "Signatures," "earmarks" and "unifications" are constellations that are prominent characteristics of a particular musical style. Furthermore, they may indicate distinctive changes in the musical sequence or serve for the internal structuring of a composition. "Signatures" are musical phrases that usually consist of characteristic melodic, harmonic and rhythmic components and often occur several times in a composition, usually in a modified form. EMI reveals "signatures" in the corpus using variably configurable pattern-matching processes, transforms them, if necessary, and also applies them in the process of generation in suitable positions within the composition. Since "signatures" may occur in different concrete musical shapes, EMI allows for their recognition and treatment within certain tolerances, meaning that, for example, a particular musical constellation on different scale steps or also in different rhythmic form may still be recognized as a consistent motif. A way to control the parameters of the pattern matching algorithms is given by the search for a particular number of "signatures" as it is typical of a composition of the respective style. The tolerance limits for identifying a signature are in this case extended until the desired number of "signatures" has been found. Furthermore, EMI allows for a corresponding manipulation of existing "signatures" of the corpus in the form of voice exchanges, different possibilities of pitch transposition, rhythmic refiguring and the like during the process of generation. "Earmarks" are characteristic movements that indicate the end or the beginning of a new formal segment of a composition. The consistent use of "earmarks" in recombination allows for a style-compliant segmentation of the material by means of musical signals such as particular cadence movements or trills. "Unifications," finally, are musical configurations whose structure is important for the internal structuring of a composition and therefore relate only to formal

[5] Meaning an expression which cannot be substituted anymore, cf. chapter 4.

[6] For "preparation," "antecedent," "statement," "consequent" as a path in the direction of the root of the graph which shows the formal functions of the musical terminal on different hierarchical levels.

elements of a single work. These patterns enable, for example, the favorable placing of a significant formal segment.

These characteristic movements are treated separately by EMI in the recombination process in order to maintain their structural integrity. Because different structural variants of, for example, motifs are recognized by EMI by means of pattern matching algorithms, they may for the application in recombination also be subjected to adequate transformations: “As the second movement started, I heard a very striking chromatically descending eight-note motive in midrange, then moments later heard the same motive way up high on the keyboard, then once again a few notes lower, [. . .] These widely spread entries gave an amazing feeling of coherence to the music [. . .] Astonished, I asked Dave what was going on and he replied, ‘Well, somewhere in one of the input movements on which this movement is drawing, there must be some motive – totally different from this motive, of course! – that occurs four times in rapid succession with exactly these same timing displacements and pitch displacements’ [. . .]”.[7] Figure 5.5 shows shows the beginning of a fugue generated by EMI based on the corpus of the fugues from J. S. Bach’s “Well-Tempered Clavier.”

5.2 Petri Nets

Petri nets[8] are a special type of transition network that is used for the simulation of event-controlled processes and are represented by bipartite graphs. Nodes may consist of data, conditions and states (*places*) or actions (*transitions*). Transitions process data from places and store it in new places. The structure of the net results from the *flow relation* which relates particular places (generally represented by circles) and transitions (generally represented by rectangles) to each other by means of directed edges. The current state of the system is indicated by *tokens* that are distributed at specific places. After the net has been initialized through the *marking* of particular places, transitions may start to act by a process referred to as *firing*. When a transition fires, it takes the tokens from its input places and puts them on its output places; in other words, information is taken from places, processed and placed at other places. Figure 5.6 (top) shows the scheme of a calculation process by means of a simple Petri net and the chronological order of the markings (figure 5.6, bottom).

[7] Douglas Hofstadter in a conversation with David Cope; in [6, p. 50].

[8] In an originally simple form also called *condition nets* or *event nets*; developed as a mathematical representation of distributed systems in the 1960s by Carl Adam Petri and first introduced in his doctoral thesis; cf. [10].

Fig. 5.5 Beginning of a fugue generated by EMI. Example kindly provided by David Cope.

5.2.1 Petri Nets in Algorithmic Composition

Goffredo Haus and Alberto Sametti [7] developed ScoreSynth, a system of algorithmic composition that enables the processing of musical information with Petri nets. By means of interconnecting "music objects" (the places) with some transforming functions (the transitions), ScoreSynth can generate and manipulate control data in the form of MIDI values in different ways. The "music objects" consist of sequences of notes with associated information on pitch, duration, velocity and MIDI channel; the transitions enable manipulation of them by crescendo, decrescendo, crab, different possibilities of transition and the like. Because in a traditional Petri net a temporal structuring of sequences is not encoded (since the transitions fire in the moment they are connected with a marked place on the input side), the places are equipped with a counter which enables access to the information of the respective "music object" only after a certain period of time. For the programming of ScoreSynth, a special syntax is developed; furthermore, the possible application possibilities of

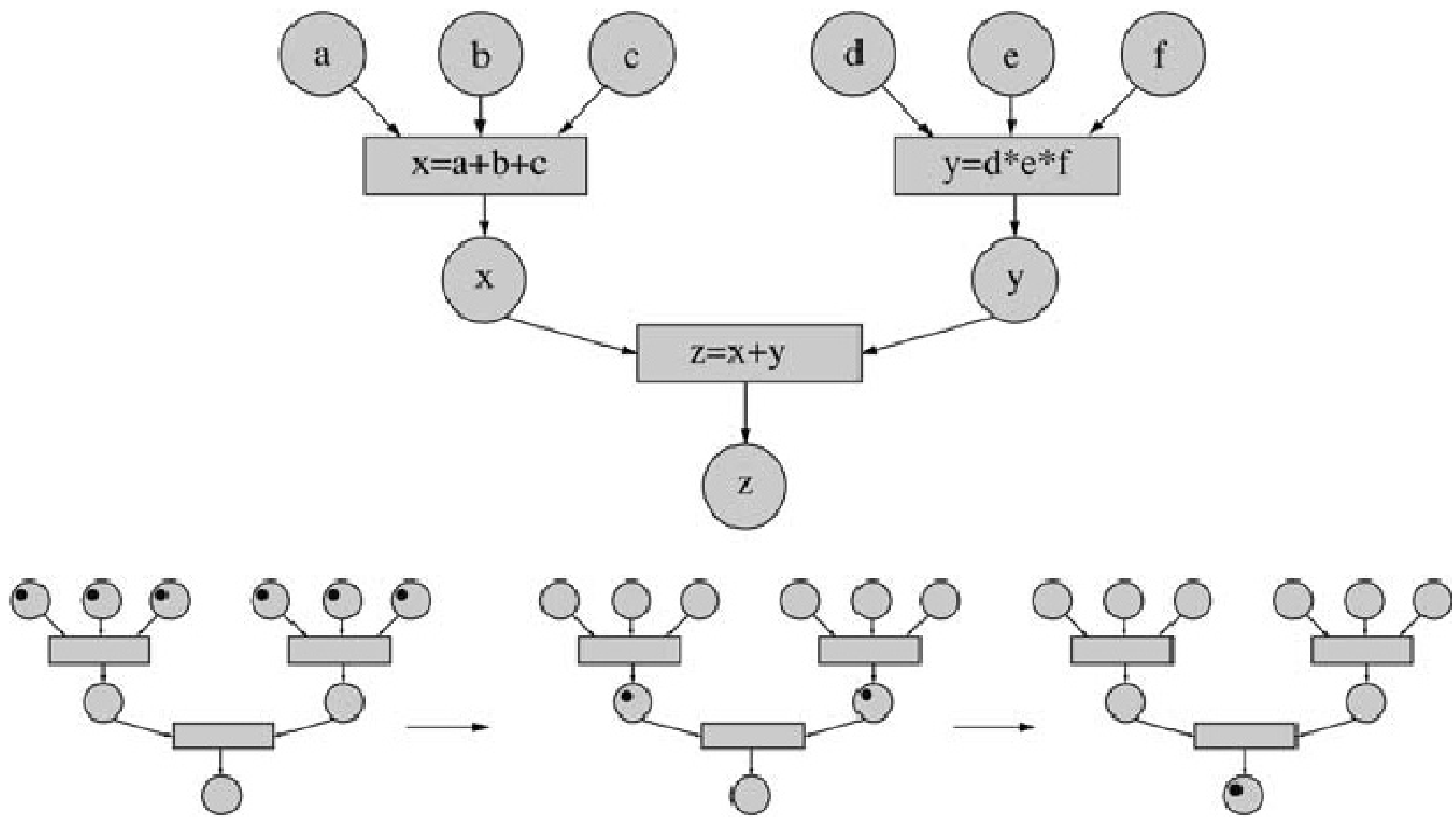

Fig. 5.6 Simple Petri net for arithmetic operations.

the software are extended by considering subnets and recursive net connections that may be restricted by a parameter controlling the number of recursions.

An interesting application of Petri nets in algorithmic composition is described by Douglas Lyon [9] who uses this formalism for modeling Markov chains of different order. The transition matrix of a Markov model is represented in a Petri net by different probabilities of weighted edges – an advantage of this approach shows in the fact that in the Petri net only transition probabilities $p_n \neq 0$ need to be processed [9, p. 19ff].

5.3 Synopsis

For the advantages and disadvantages of transition networks regarding tasks of algorithmic composition, in general the same principles apply as for generative grammars. The augmented transition network, for example, equals the type-0 grammar in terms of its expressive power. In contrast to genetic algorithms or cellular automata, for example, whose strong points become apparent in the realization of very specific compositional concepts, transition networks may be used in a broad field of musical structure genesis. It is exactly this aspect of the "universal" applicability of systems like these that makes them especially suitable for the development of algorithmic composition systems which enable the formulation of different compositional strategies in the sense of a programming language or the realization of a complex system design, as in EMI. An essential difference between a TN and generative grammars, and also Lindenmayer systems, may be seen in the representation of musical information within a graph. This difference also shows in the design of the user interface of computer music systems where– in contrast to a compact formulation of instructions within a text-based system – visual objects are manipulated. The graph representation of transition networks finds its parallels in computer music systems such as MAX,[9] PureData[10] or OpenMusic[11] that enable the manipulation of the musical information within a graphically interconnected structure.

References

1. Cope D (1987) An expert system for computer-assisted composition. Computer Music Journal 11/4
2. Cope D (1991) Computers and musical style. Oxford University Press, Oxford. ISBN 0-19-816274-X
3. Cope D (1992) On algorithmic representation of musical style. In: Balaban M, Ebcioglu K, Laske O (eds) (1992) Understanding music with AI. AAAI Press/MIT, Cambridge, Mass. ISBN 0262-52170-9
4. Cope D (1996) Experiments in musical intelligence. A-R Editions, Madison, Wis. ISBN 0-89579-314-8
5. Cope D (2000) The algorithmic composer. AR-Editions, Madison, Wis. ISBN 0-89579-454-3
6. Cope D (2001) Virtual music: computer synthesis of musical style. MIT Press, Cambridge, Mass. ISBN 0-262-03283-X
7. Haus G, Sametti A (1991) SCORESYNTH: a system for the synthesis of music scores based on Petri nets and a music algebra. IEEE Computer, 24/7, July, 1991
8. Hofstadter D (1979) Goedel, Escher, Bach: an eternal golden braid. Basic Books, New York. ISBN 0465026850
9. Lyon D (1995) Using stochastic Petri nets for realtime nth-order stochastic composition. In: Computer Music Journal 19/4, 1995
10. Petri CA (1962) Kommunikation mit Automaten. Schriften des Instituts für angewandte Mathematik, Bonn
11. Woods WA (1970) Transition network grammars for natural language analysis. Communications of the ACM, October 1970

[9] See: http://www.cycling74.com/.

[10] See: http://puredata.info/.

[11] A software specialized in algorithmic composition, developed by IRCAM: http://ircam.fr/; also see chapter 10.