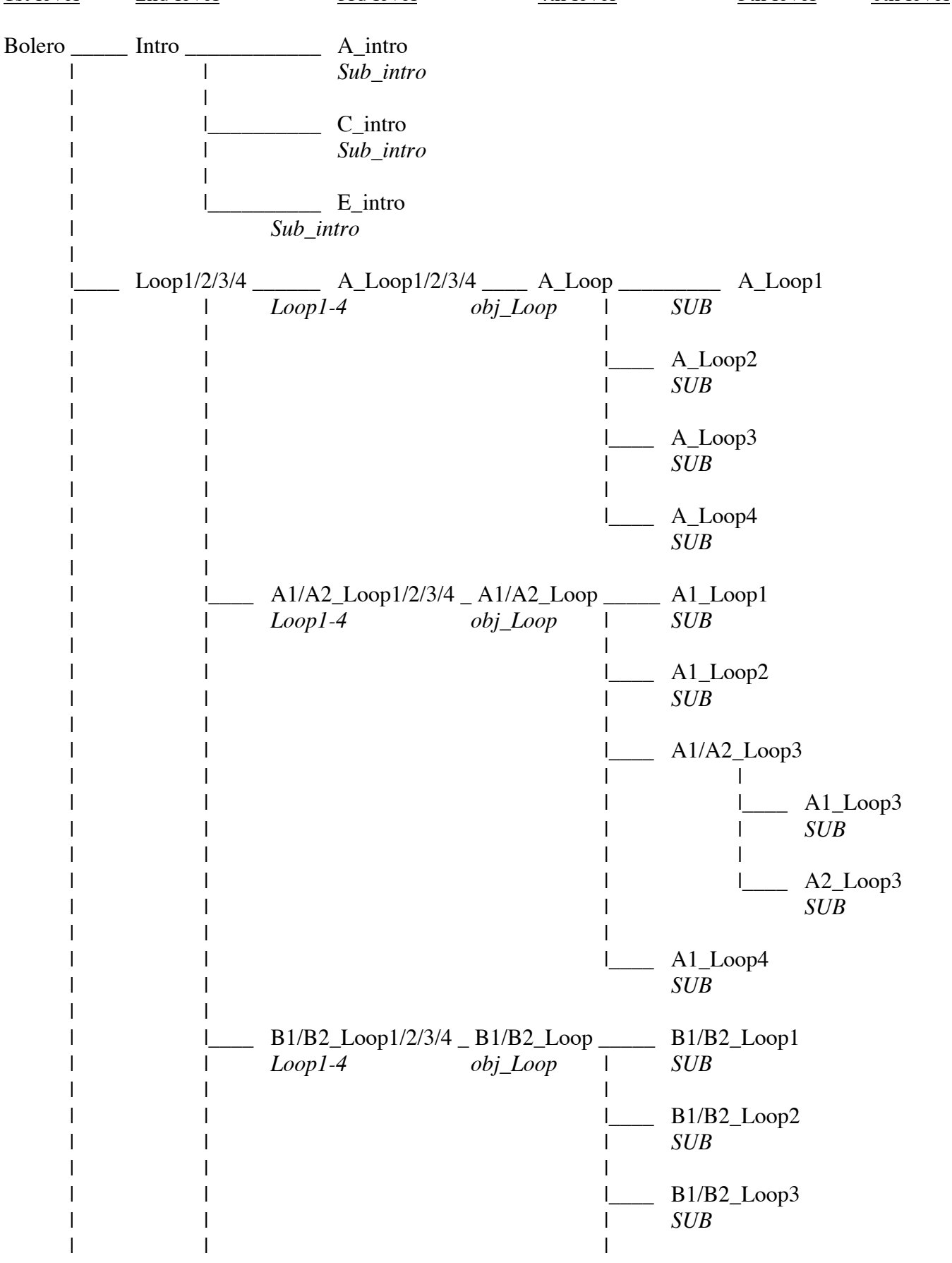
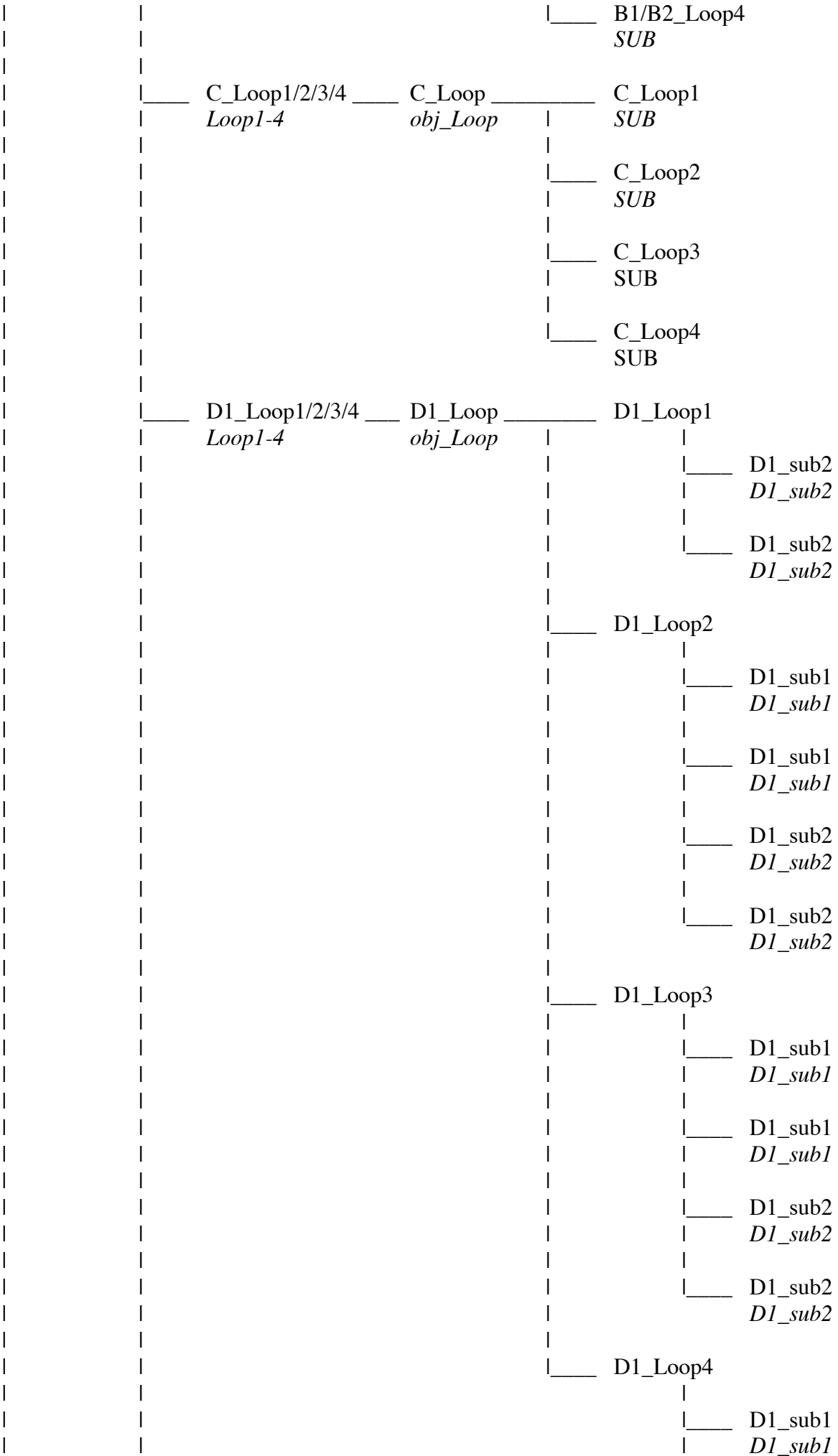
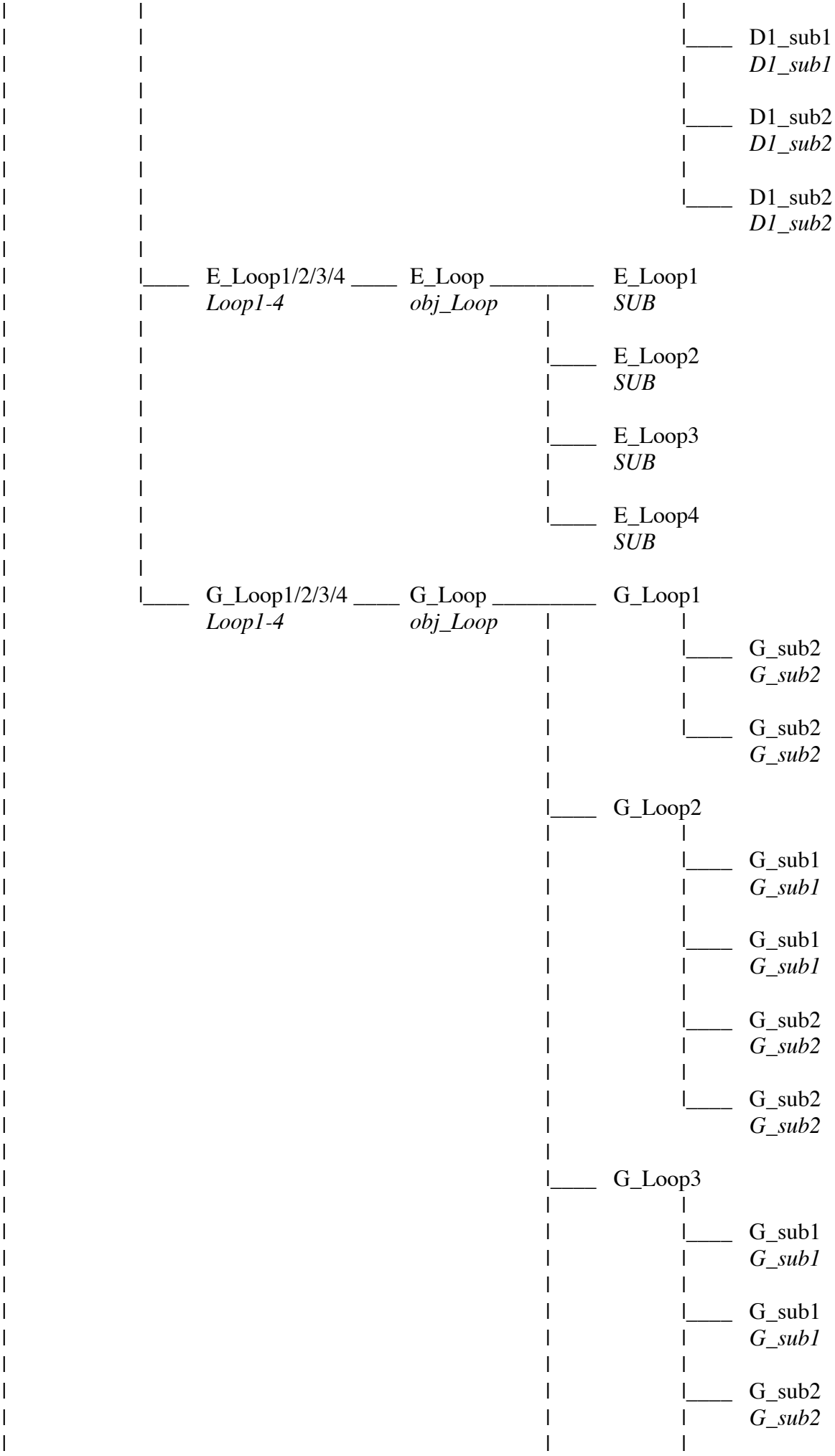
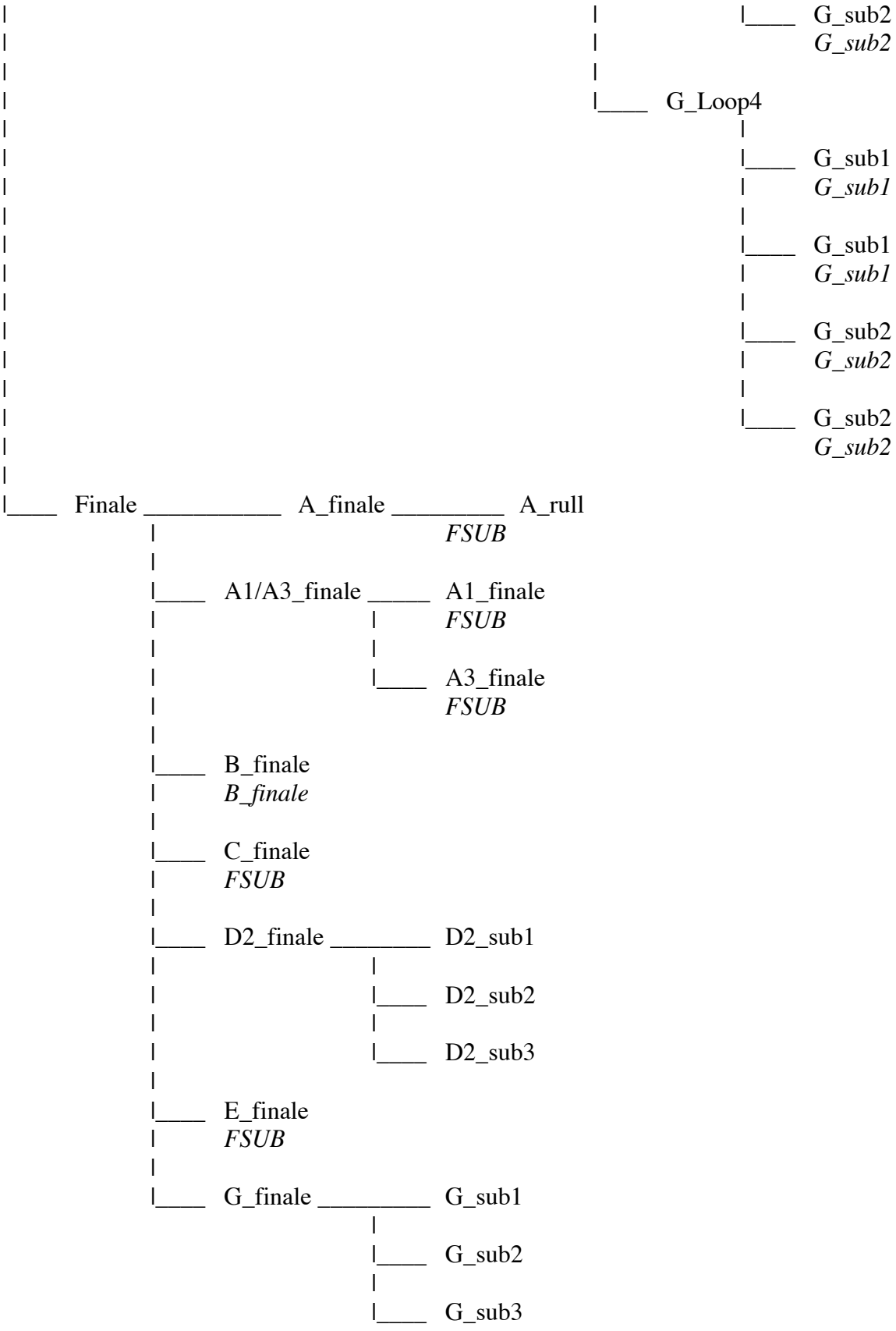
3.4. Model Architecture

The model we have defined is a particular one with respect to the many models we can implement by PNs; it has the following hierarchical architecture made up of PNs' morphisms and macro invokings (we have used italics to specify macro nets' identifiers:




B1/B2_Loop4
SUB
C_Loop1/2/3/4
Loop1-4
C_Loop
obj_Loop
C_Loop1
SUB
C_Loop2
SUB
C_Loop3
SUB
C_Loop4
SUB
D1_Loop1/2/3/4
Loop1-4
D1_Loop
obj_Loop
D1_Loop1
D1_sub2
D1_sub2
D1_sub2
D1_sub2
D1_Loop2
D1_sub1
D1_sub1
D1_sub1
D1_sub1
D1_sub2
D1_sub2
D1_sub2
D1_sub2
D1_Loop3
D1_sub1
D1_sub1
D1_sub1
D1_sub1
D1_sub2
D1_sub2
D1_sub2
D1_sub2
D1_Loop4
D1_sub1
D1_sub1


D1_sub1
D1_sub1
D1_sub2
D1_sub2
D1_sub2
D1_sub2
E_Loop1/2/3/4
Loop1-4
E_Loop
obj_Loop
E_Loop1
SUB
E_Loop2
SUB
E_Loop3
SUB
E_Loop4
SUB
G_Loop1/2/3/4
Loop1-4
G_Loop
obj_Loop
G_Loop1
G_sub2
G_sub2
G_sub2
G_sub2
G_Loop2
G_sub1
G_sub1
G_sub1
G_sub1
G_sub2
G_sub2
G_sub2
G_sub2
G_Loop3
G_sub1
G_sub1
G_sub1
G_sub1
G_sub2
G_sub2



In other words, all the model can be considered as an only whole net which is represented by means of a hierarchical graphic language; in fact, SCORESYNTH (the PNs executer we have used) expands both all the morphisms and all the PNs' macro invokings before executing the model. In the following paragraphs we summarize the characteristics of our model.

Some global numbers about the model seem to be interesting with respect to complexity and quantities of information within the model itself: there are 11 macro nets, 72 macro invokings, 19 explicitly defined (not macro) nets, 6 levels of abstraction, 15 music objects.

Let's begin to observe the model at the most abstract level. In Fig. 17 we show how the Bolero score can be considered as a three portions piece: the first two bars (**Intro**), the central portion that is made up of four cyclic structures each one 72-bars long (**Loop1/2/3/4**) and the ending portion (**Finale**).

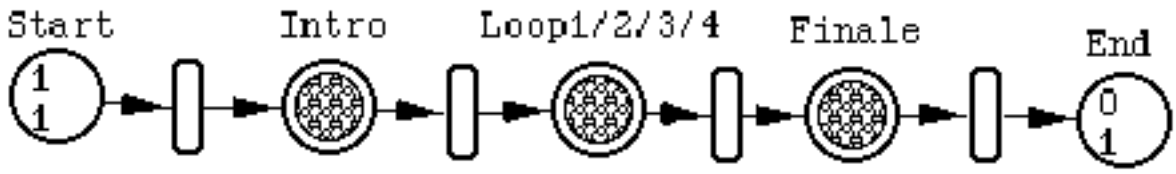


Figure 17 The most abstract net of our model.

We describe the model starting from this very simple net and the family of music objects of which we have spoken in § 2.

3.4.1. Model Conventions

Before going deeper, we have to mention some relevant conventions we have adopted within the development of the model; they concern dynamics, MIDI channels and timbre textures.

The algorithms of the SCORESYNTH program allow to control dynamic changes of music objects; the dynamics of Bolero grow all over the development of the piece starting from the "pianissimo" up to reach the "fortissimo". MIDI key velocity codes allow to control dynamics by a numeric range (from 0 up to 127); so, we have defined the following table of correspondence between score specifications and MIDI codes:

- pp	18	(pianissimo)
- p	28	(piano)
- mp	38	(mezzopiano)
- mf	48	(mezzoforte)
- f	58	(forte)
- ff	68	(fortissimo).

Our first approach about instrumentation was to assign an audio process to every music instrument of the Bolero score, but the analysis of music objects' life within the piece has revealed that it is not the right approach. In fact, we have seen that Ravel uses traditional instruments as if they were samplers or, better, complex wave oscillators in order to do a kind of additive synthesis. Therefore, we have defined another kind of approach in which every music object has its own audio process in which timbre is the complex result corrisponding to the orchestral texture designed by Ravel. The implementation of this approach is made by MIDI channels and samplers: a MIDI channel is used for one only music object at a time and every timbre changing has its corresponding sample change by means of suitable MIDI exclusive commands. This approach is particularly clear from the conceptual point of view and makes the score highly transparent to observers: the life of music objects is now explicitly described while the traditional score hides their life due to the often changing distributions of music objects within different instrumental parts.

The implementation table of MIDI channels with respect to music objects is given below; sometime we have more than one music object associated to a channel: it means that only one music object at a time is alive in that channel:

- **A**, **Ares**	1
- **A1**	2
- **A2**, **A3**	3
- **B1**, **B2**, **B3**, **B4**, **Bres**	4
- **C**	5
- **D1**, **D2**	6

- **E** 7
- **G** 8

3.4.2. The "Intro" SubModel

The **Intro** subnet (see Fig. 18) has three places (**A_intro**, **C_intro** and **E_intro**) devoted to the description of the behaviour of objects **A**, **C** and **E** respectively, in the first two bars of the score. All these places are the abstraction of the same macro net in Fig. 19 which is actualized by the three modifier lists shown in the three frames of Fig. 18. The algorithm applied to the original objects is

L:1,$,18

which assign a pianissimo dynamic value to all notes.

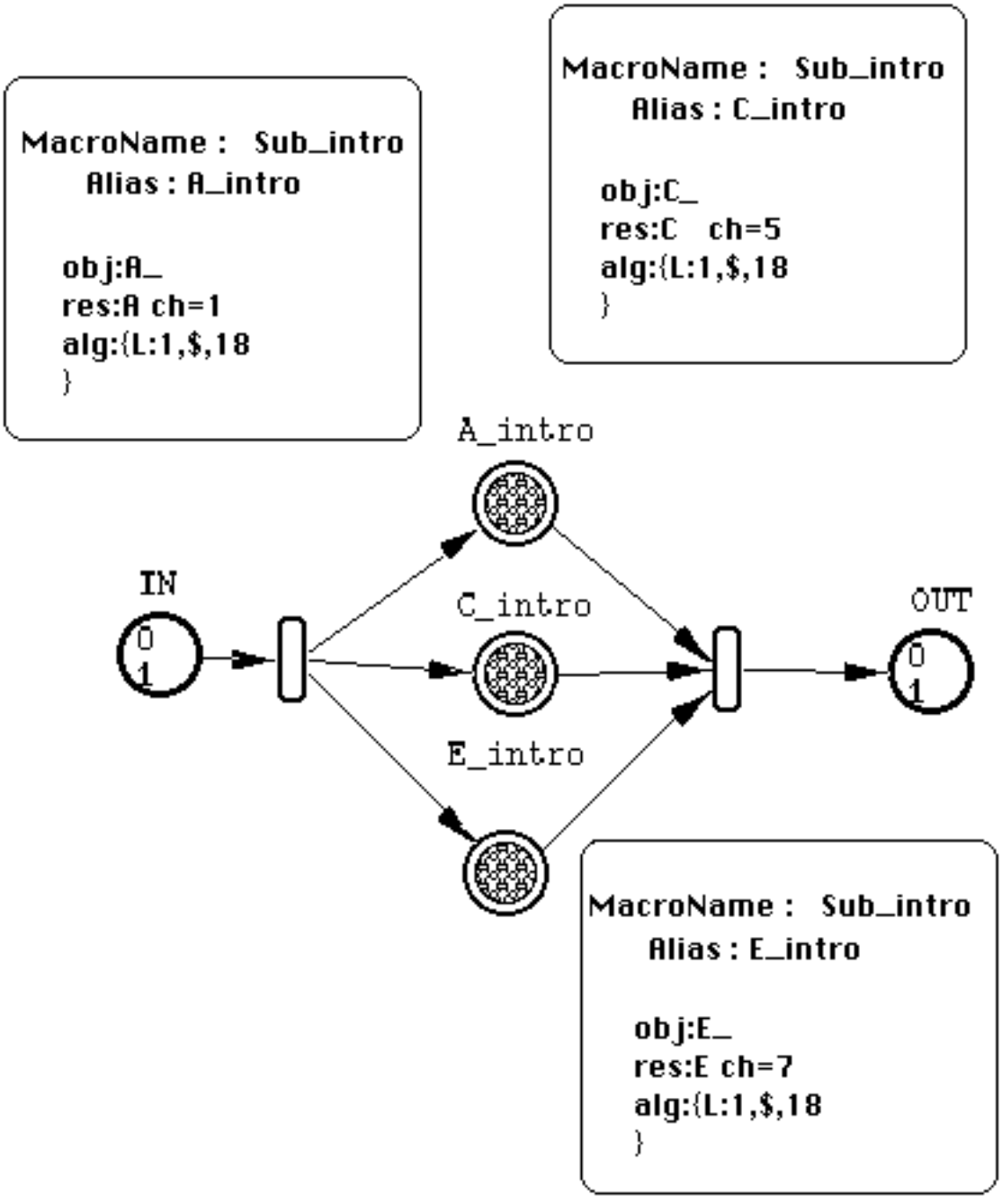


Figure 18 The **Intro** subnet.

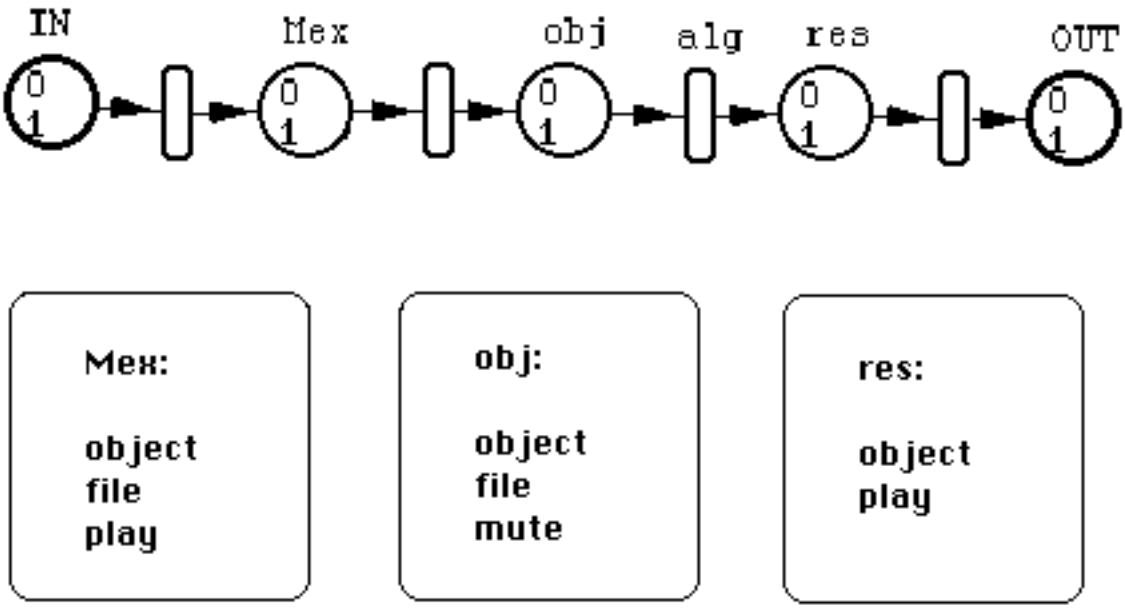


Figure 19 Macro net **Sub_intro** with related place attributes.

The **Sub_intro** macro is only used at the beginning of the score. The **Mex** place has a MIDI exclusive command associated from file; the **obj** place has objects **A**, **C** and **E** associated in turn: they are defined in the modifier lists associated to the **A_intro**, **C_intro** and **E_intro** in the **Intro** subnet, respectively; they are read from files and are not sent for playing. Attributes of places with associated objects are described in the three frames of Fig. 19. Place **res** receives the object transformed (which is in turn of the **A**, **C** or **E** kind) by the algorithm associated to transition **alg** and drives it to the channel specified in the modifier lists of Fig. 18 for playing.

3.4.3. The "Loop1/2/3/4" SubModel

The central section of the model contains the real development process of Bolero. Our model has seven subnets within the **Loop1/2/3/4** to describe the four loops, as it is shown in Fig. 20. All places (with the exception of **IN** and **OUT**) call the macro net **Loop1-4**; all the seven modifier lists are described in the frames of Fig. 21.

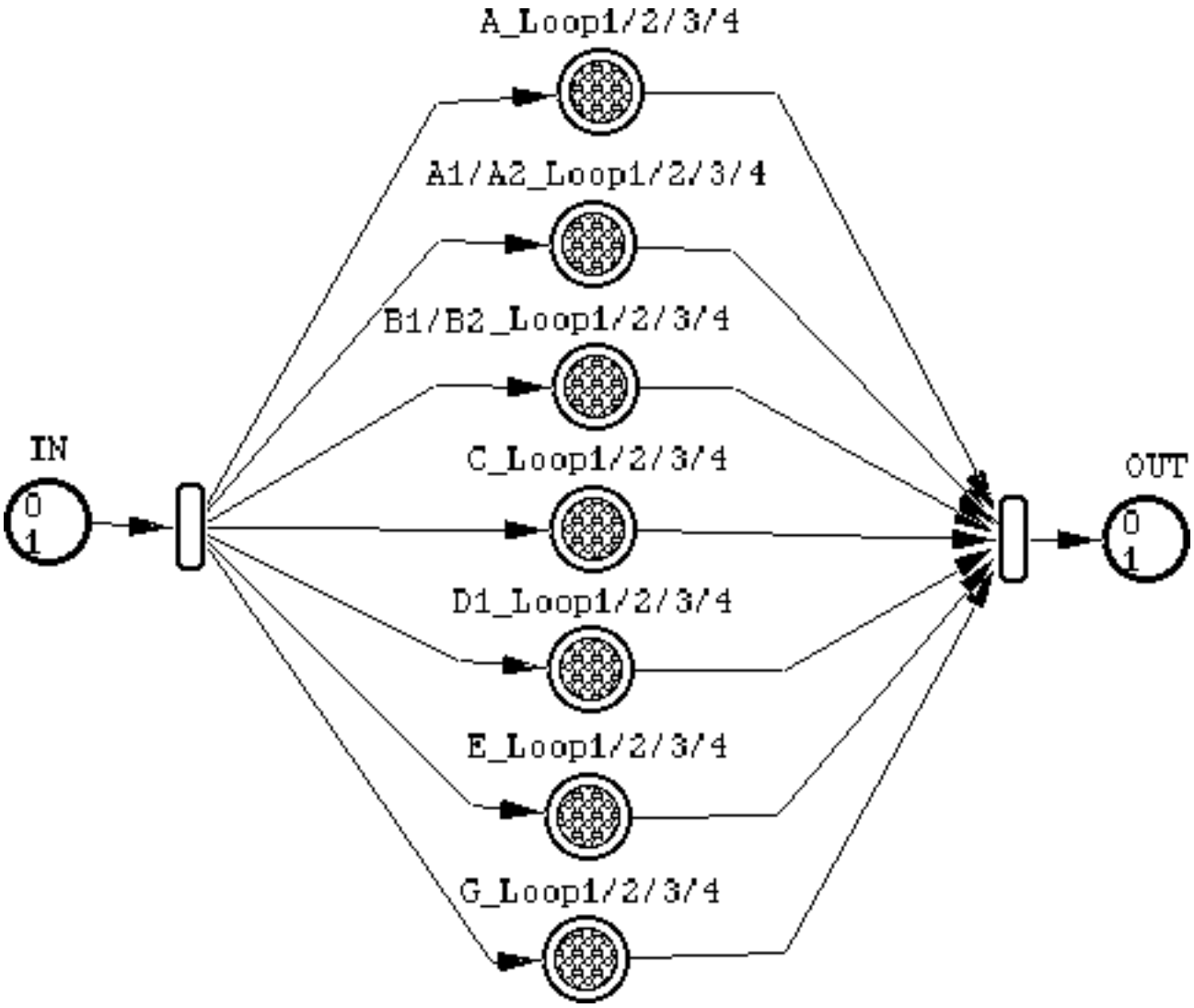


Figure 20 The central net **Loop1/2/3/4**.

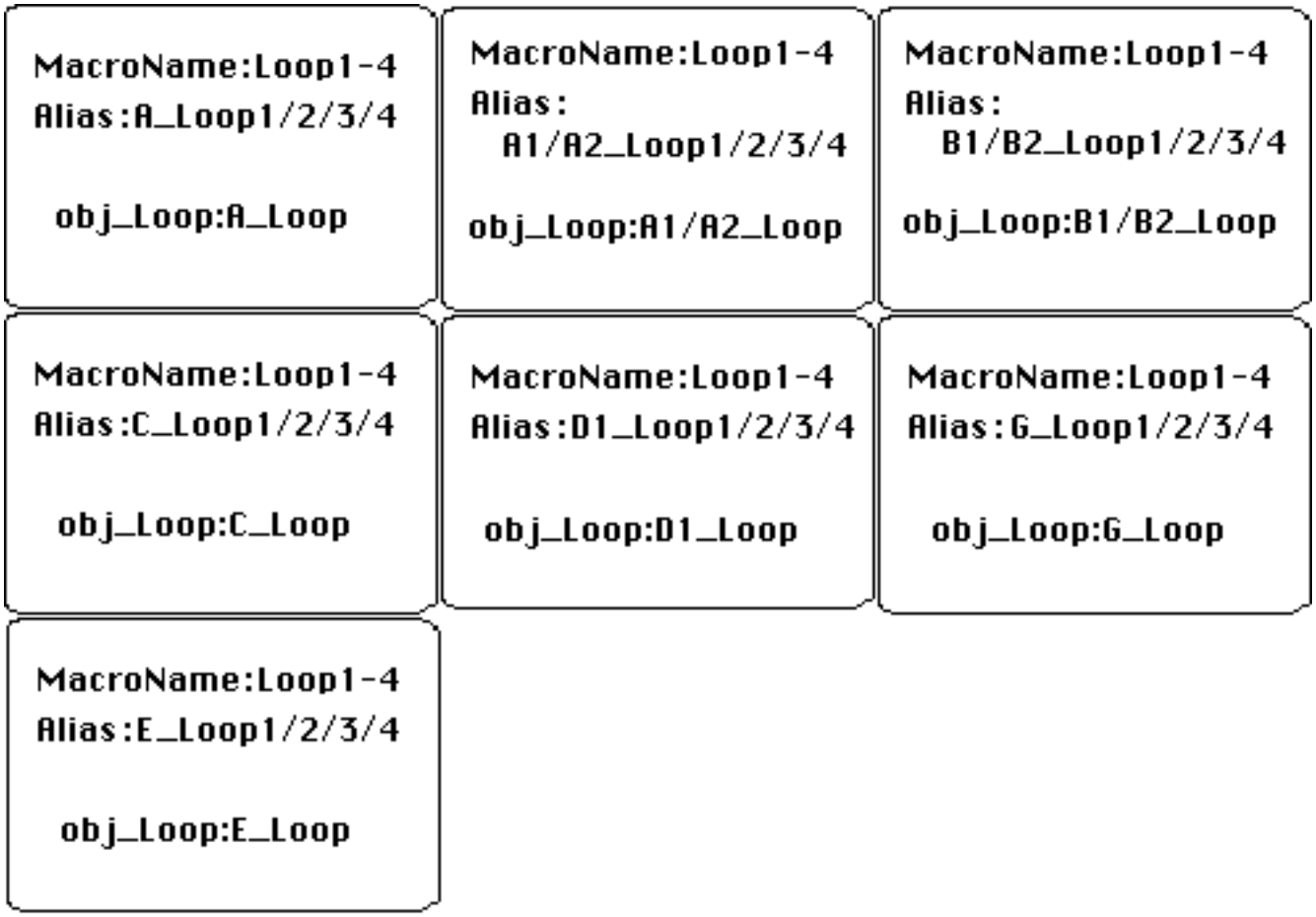


Figure 21 Modifier lists associated to the places of net **Loop1/2/3/4** to invoke macro net **Loop1-4**.

The macro net **Loop1-4** (see Fig. 22) has one of the typical net structures that are suitable for cycles' description. The subnet that describes the four loops is associated to the place **obj_Loop** once for each music object. The place **obj_Loop** receives four tokens, one for each transition firing, because the place **count** has four tokens, one for each loop structure of the original Bolero.

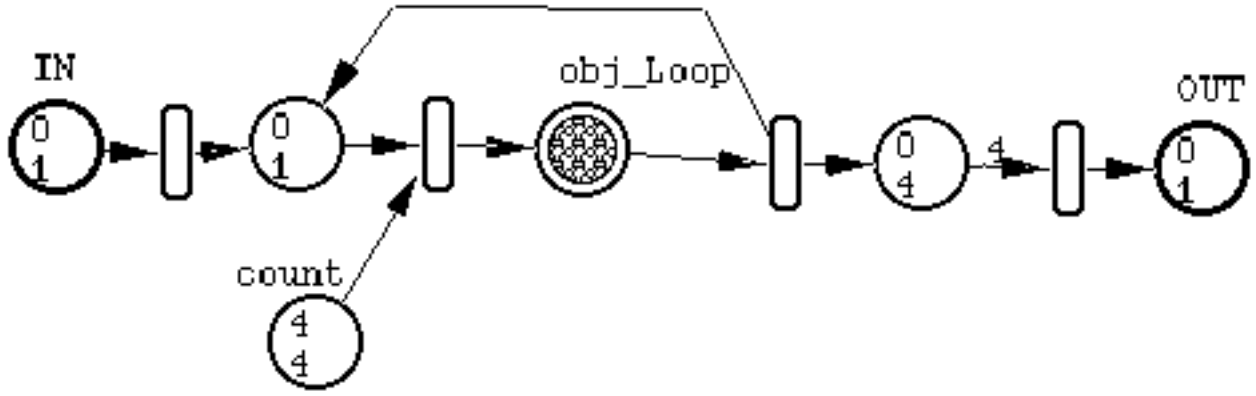


Figure 22 Macro net **Loop1-4**.

All the subnets associated to the place **obj_Loop** has the same net structure but different features depending on music objects and macro nets invoking involved; we show here the subnet version for music object **A**, while we only describe the specific invoking modifier lists associated to macro nodes for the other music objects.

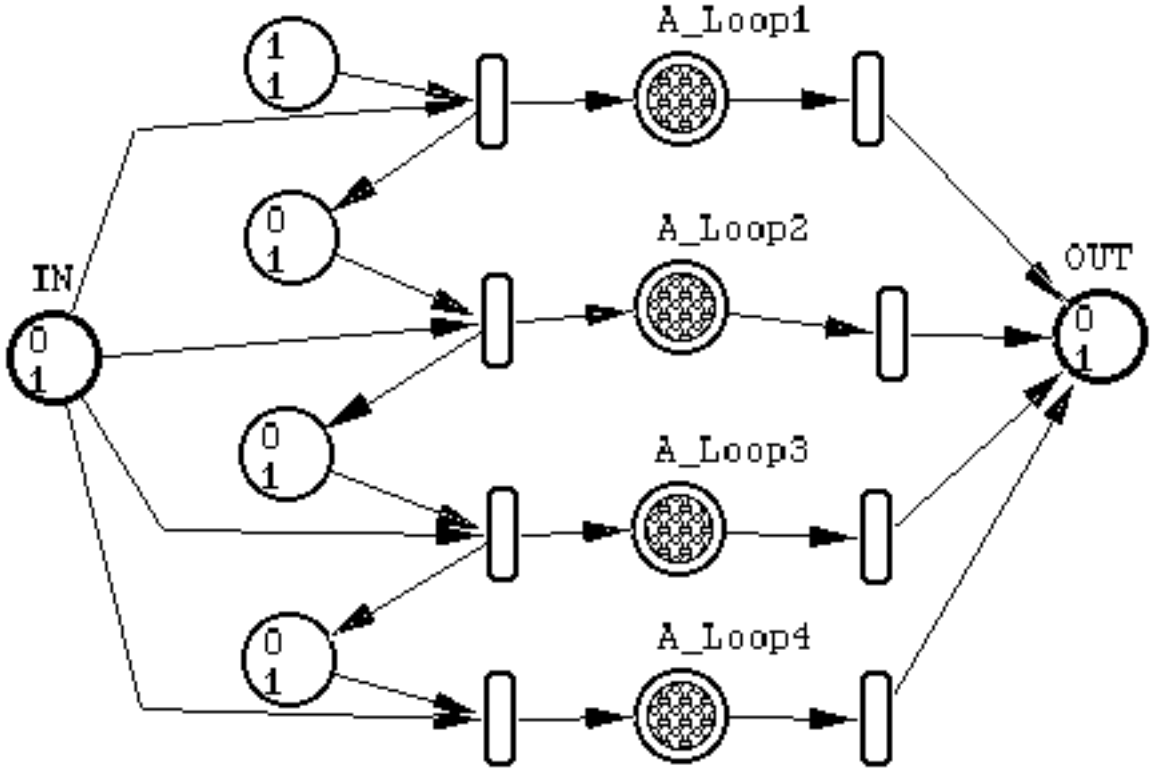


Figure 23 Subnet **A_Loop**.

The net of Fig. 23 represents a selecting structure (very similar to a *case* structure). When the first token reaches place **IN**, the only transition that can fire is the input one to place **A_Loop1**. When the second token reaches place **IN**, the state of the net has changed as it is shown in Fig. 24.

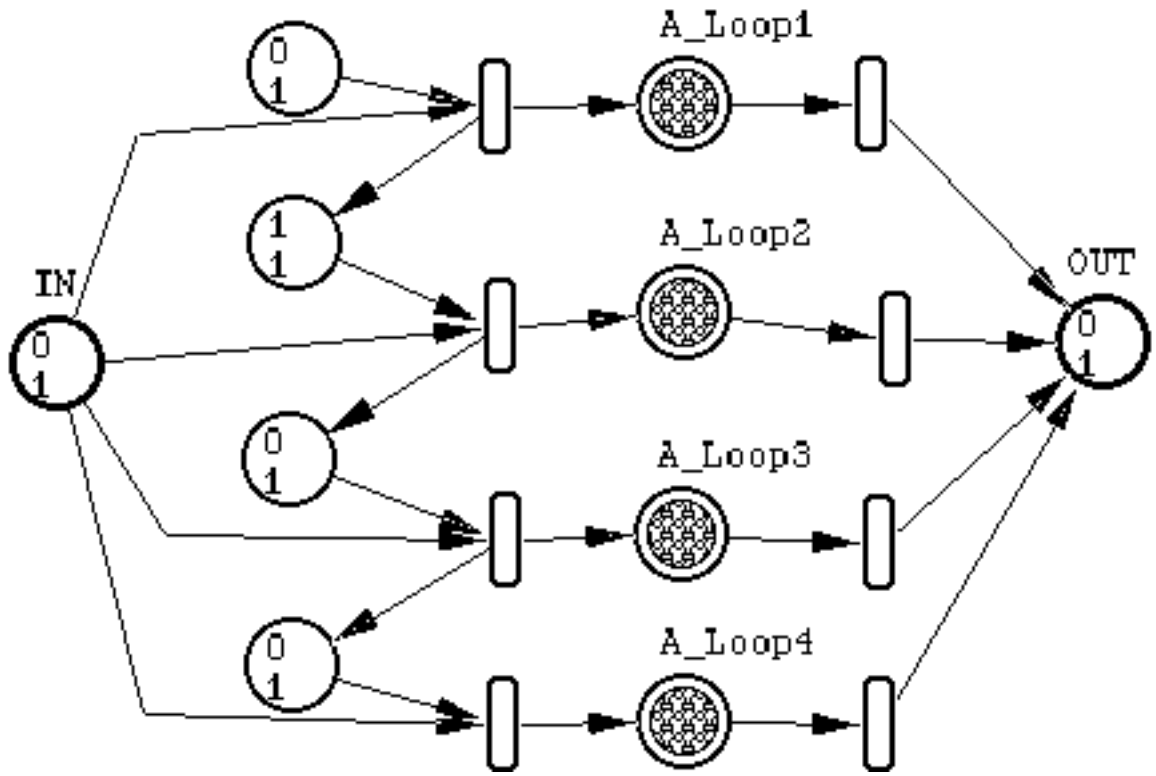


Figure 24 Net **A_Loop** at the second invoking.

In fact, the firing of the first transition within net **A_Loop1/2/3/4**, in addition to make active the output macro place **A_Loop1**, has put a token into the other output place of the transition. The token's moving is relevant because when place **IN** will receive the second token we have another transition that can fire, the

one that has place **A_Loop2** as output macro place. So, it is clear now that net **A_Loop** works for sequentially executing macro places **A_Loop1**, **A_Loop2**, **A_Loop3**, **A_Loop4**.

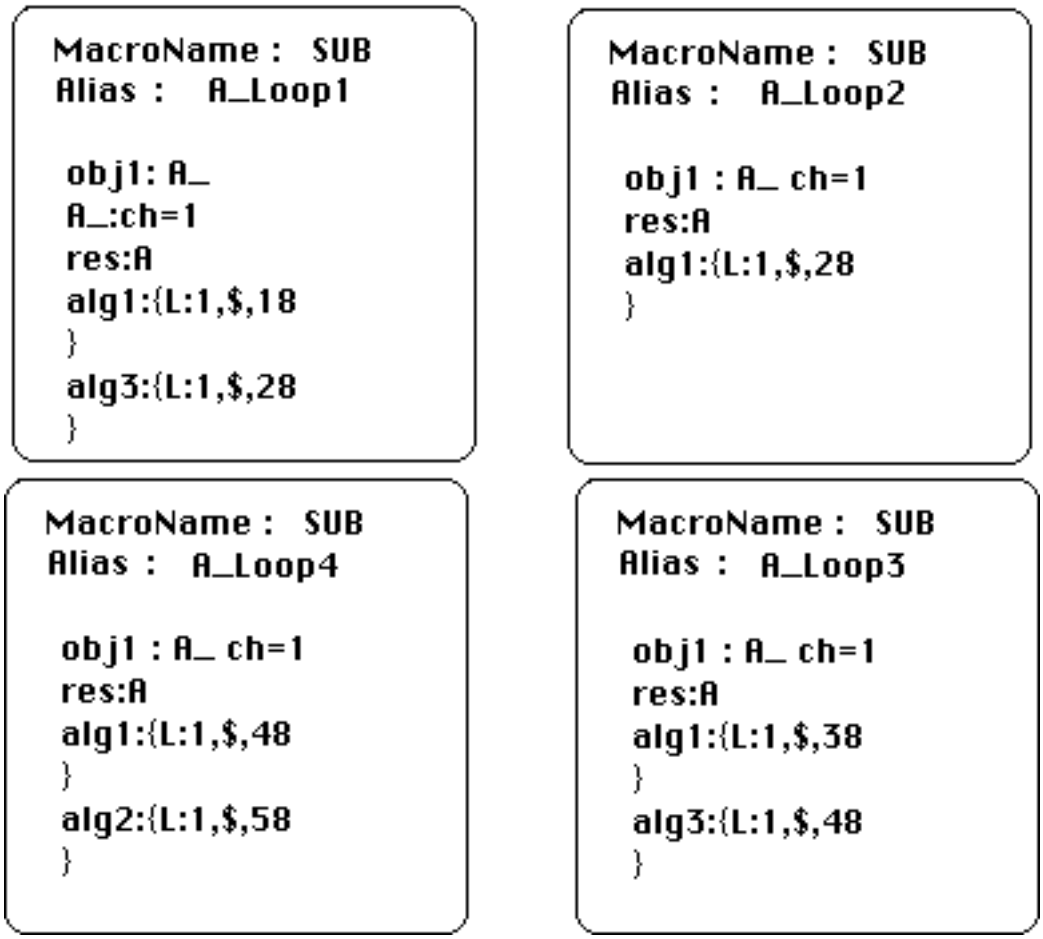


Figure 25 Invoking modifier lists associated to the places of net **A_Loop**.

Figure 25 shows the invoking modifier lists that instantiate both place and transition attributes within net **A_Loop**. Each of its macro places are refined by macro **SUB** (see Fig. 26). Music object **A** has to be played on MIDI channel 1, it is associated to place **obj1**, **A_** is the identifier of the file containing music object **A**; the algorithms associated to transitions **alg1**, **alg2**, **alg3** and **alg4** are applied on dynamics of the related music object; identifier **A** is assigned to place **res**, to which the music object produced by algorithms is associated.

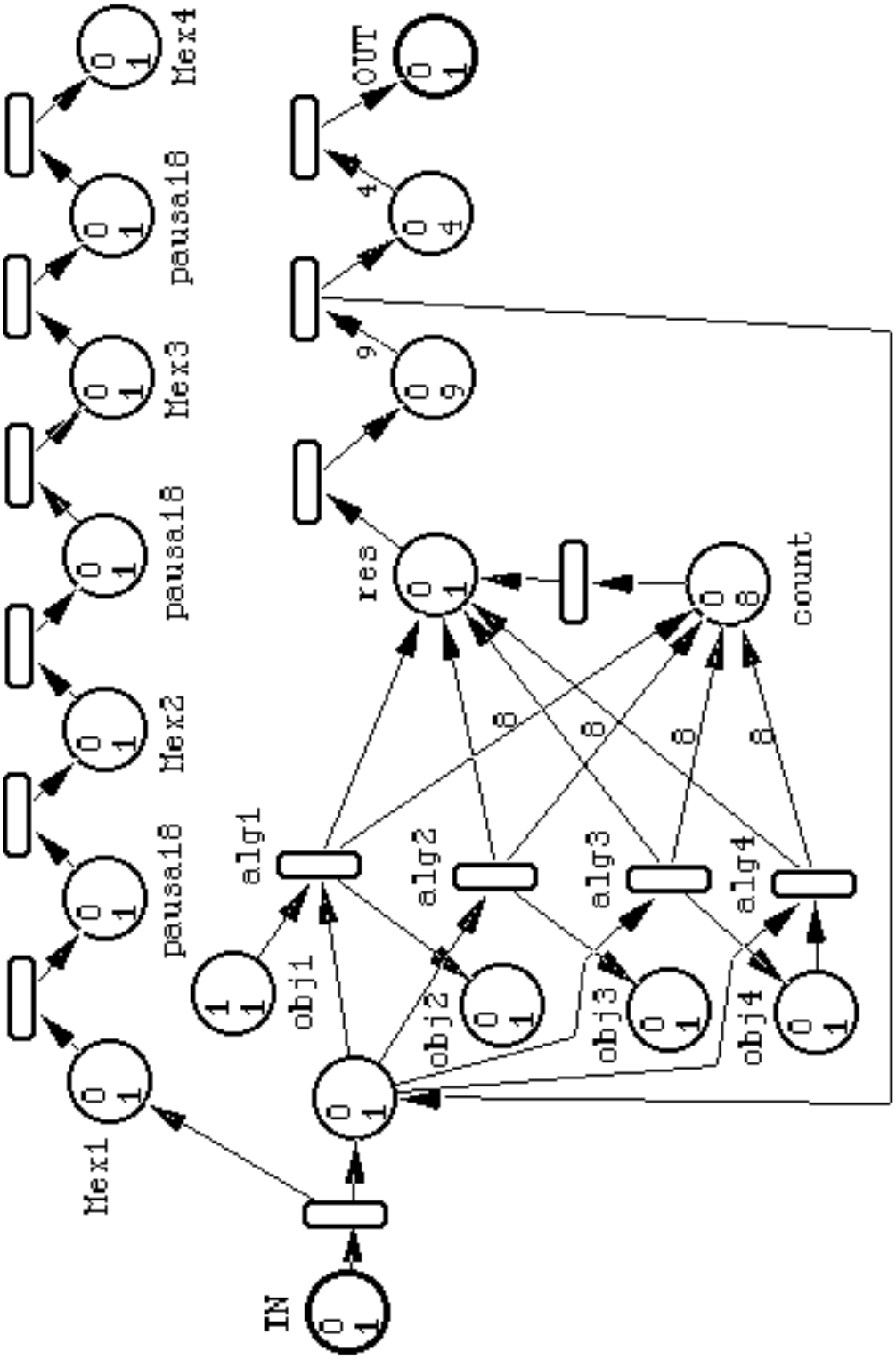


Figure 26 Macro net **SUB**.

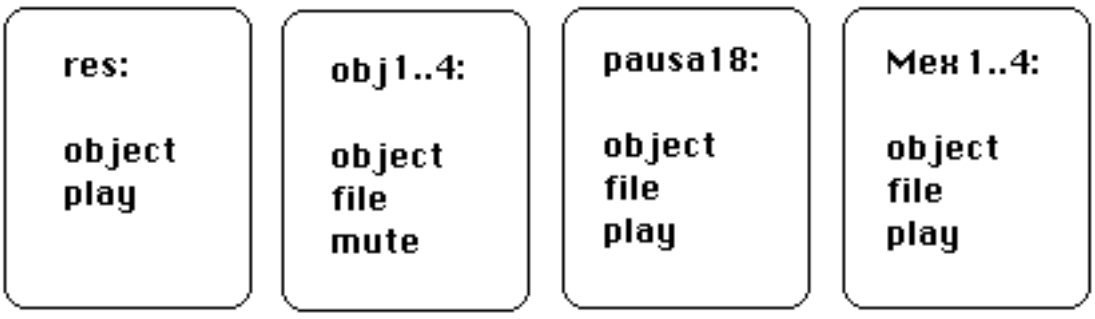


Figure 27 Attributes of places within macro net **SUB**.

In Fig. 27 the attributes of places within macro net **SUB** are shown. The macro net **SUB** is very relevant within our model: it describes all the music object processes based on two-bars-objects (i.e. the rhythmic objects) within a whole loop of Bolero; music objects, which are stored on files, are associated to places **obj1**, **obj2**, **obj3** and **obj4** that correspond to the four quarters of the loop. Place **res** is enabled to play and contains music objects produced from time to time by transitions **alg1**, **alg2**, **alg3** and **alg4** whose algorithms are applied to the music objects associated to places **obj1**, **obj2**, **obj3** and **obj4**. Place **Mex1**, **Mex2**, **Mex3** and **Mex4** have commands to control timbral textures changes associated. Places **pausa18** have a rest associated so that music objects are executed at the right moment, i.e. every nine executions of the music object associated to place **res**.

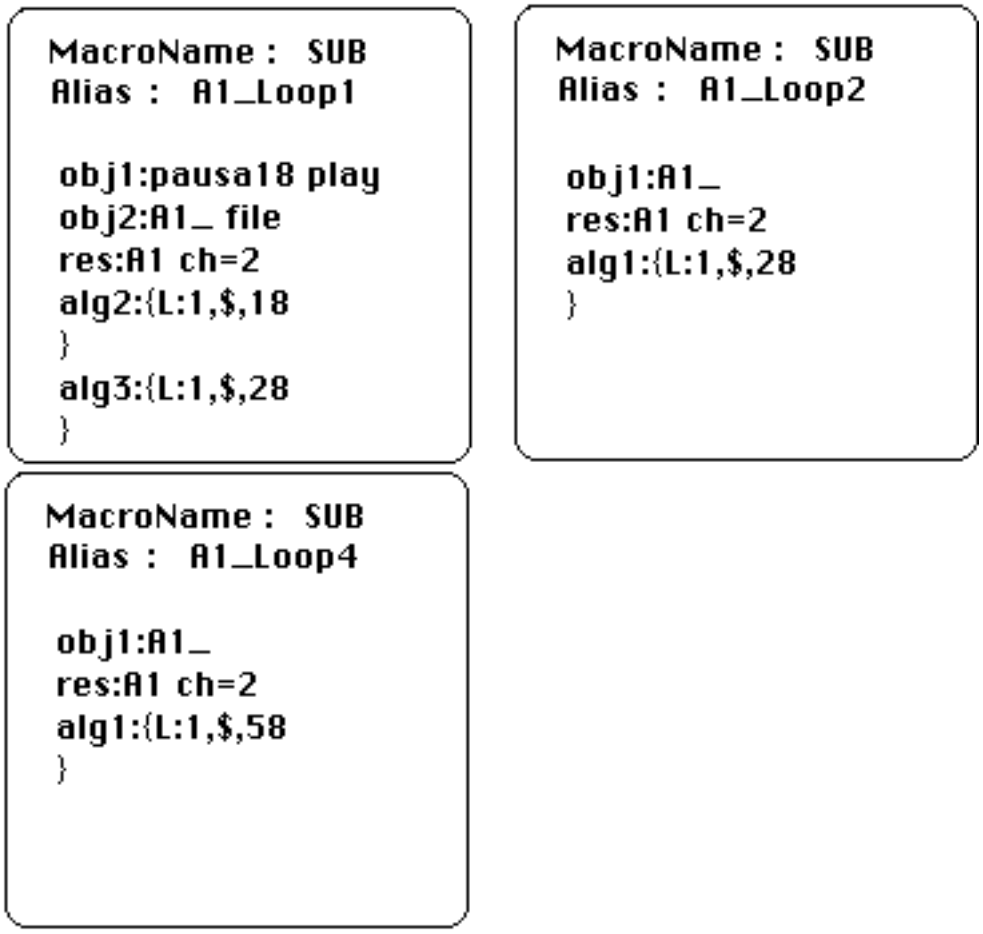
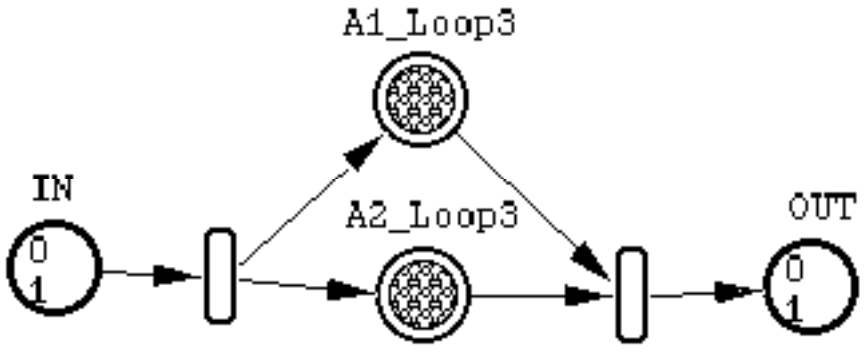


Figure 28 Invoking modifier lists of macro places within subnet **A1/A2_Loop**.

Subnet **A1/A2_Loop** has the same structure of net **A_Loop**; the only differences concern place identifiers that now become **A1_Loop1**, **A1_Loop2**, **A1/A2_Loop3** and **A1_Loop4**. Fig. 28 shows invoking modifier lists of the three macro places within the subnet; place **A1/A2_Loop3** is instead refined by the subnet shown in Fig. 29.



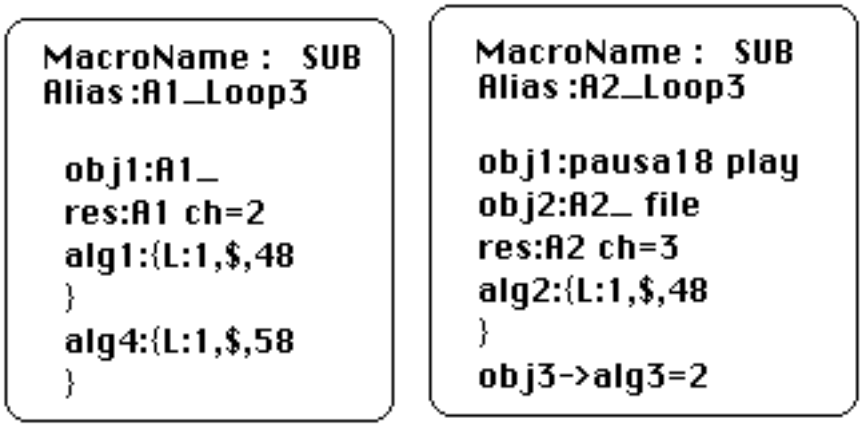


Figure 29 Subnet **A1/A2_Loop3** and invoking modifier lists associated to its macro places.

Really, the only function of subnet **A1/A2_Loop3** is to split the node which invokes it; in fact, in the third loop of Bolero, music object **A2** appears concurrently to music object **A1**. Fig. 29 shows also invoking modifier lists associated to macro places **A1_Loop3** and **A2_Loop3**.

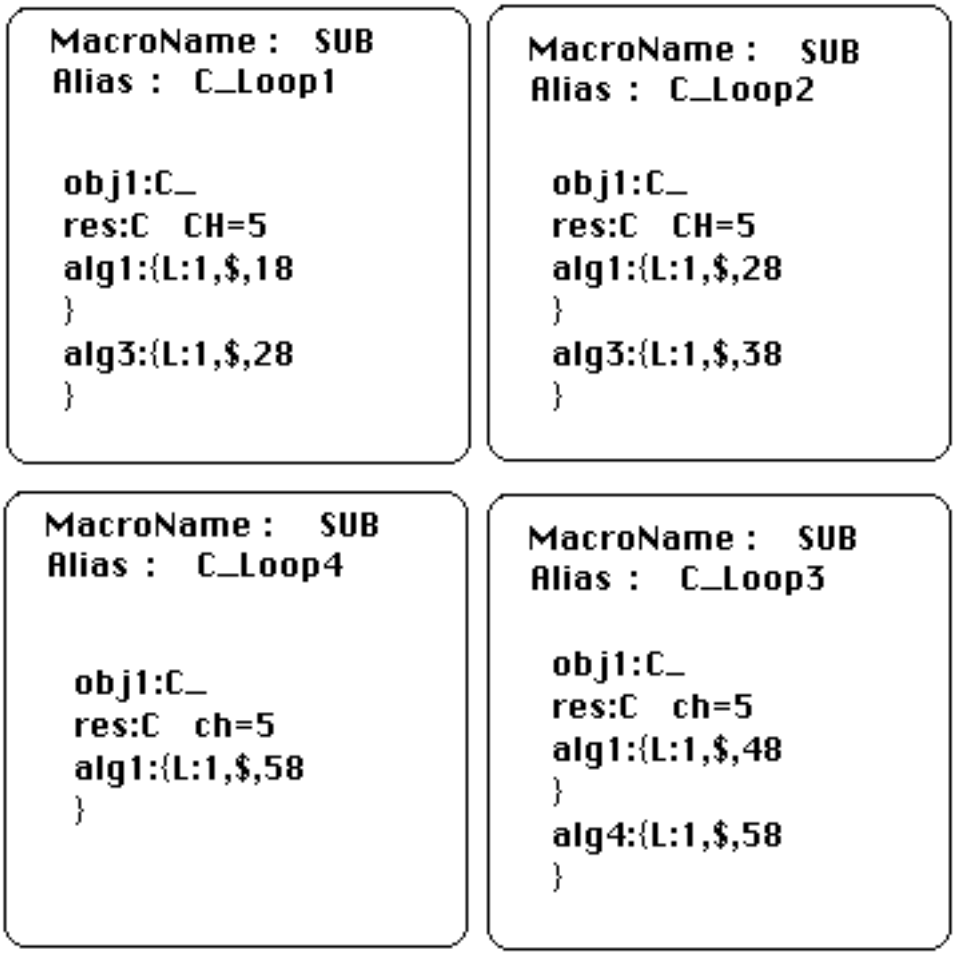


Figure 30 Invoking modifier lists associated to macro places within subnet **C_Loop**.

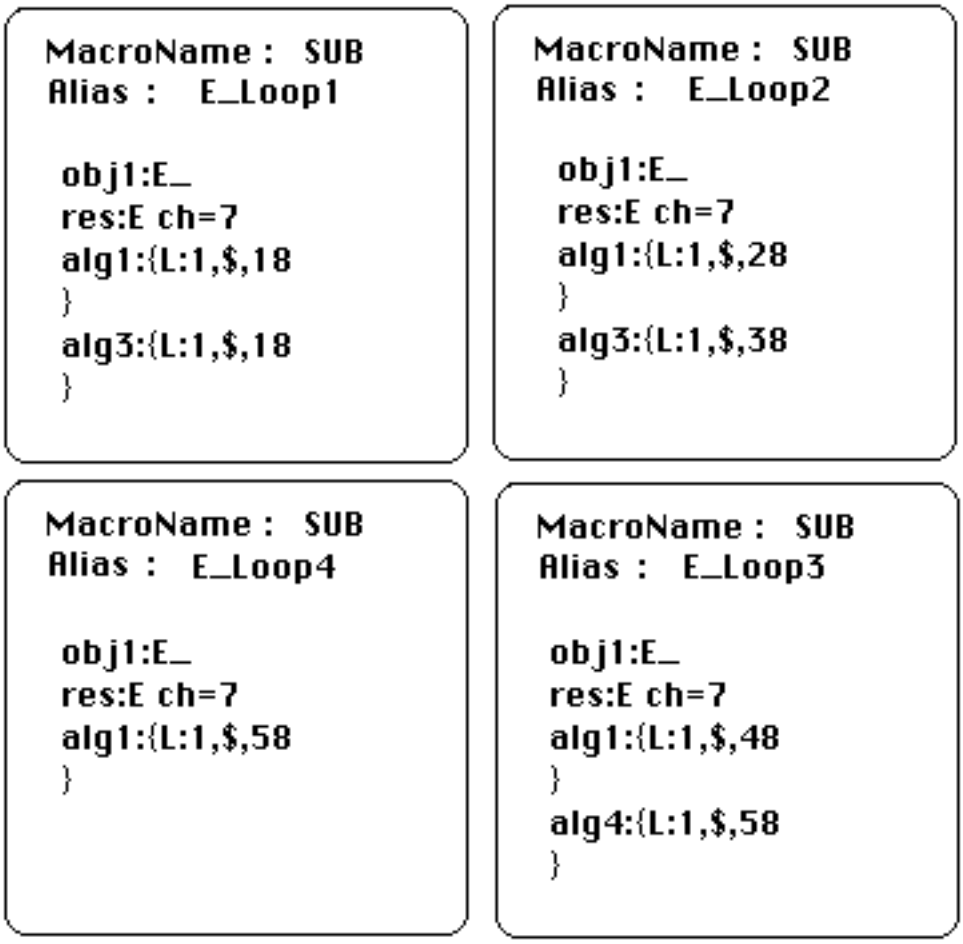


Figure 31 Invoking modifier lists associated to macro places within subnet **E_Loop**.

Figures 30 and 31 show the invoking modifier lists associated to macro places within subnets **C_Loop** and **E_Loop** respectively. So far, we have seen all the nets concerning rhythmic objects (**A**, **A1**, **A2**, **C** and **E**) along the four loops of Bolero. Now we can speak about the nets that describe the behaviour of melodic objects **B1** and **B2**, bearing in mind that subnet **B1/B2_Loop** has the same structure of subnet **A_Loop**.

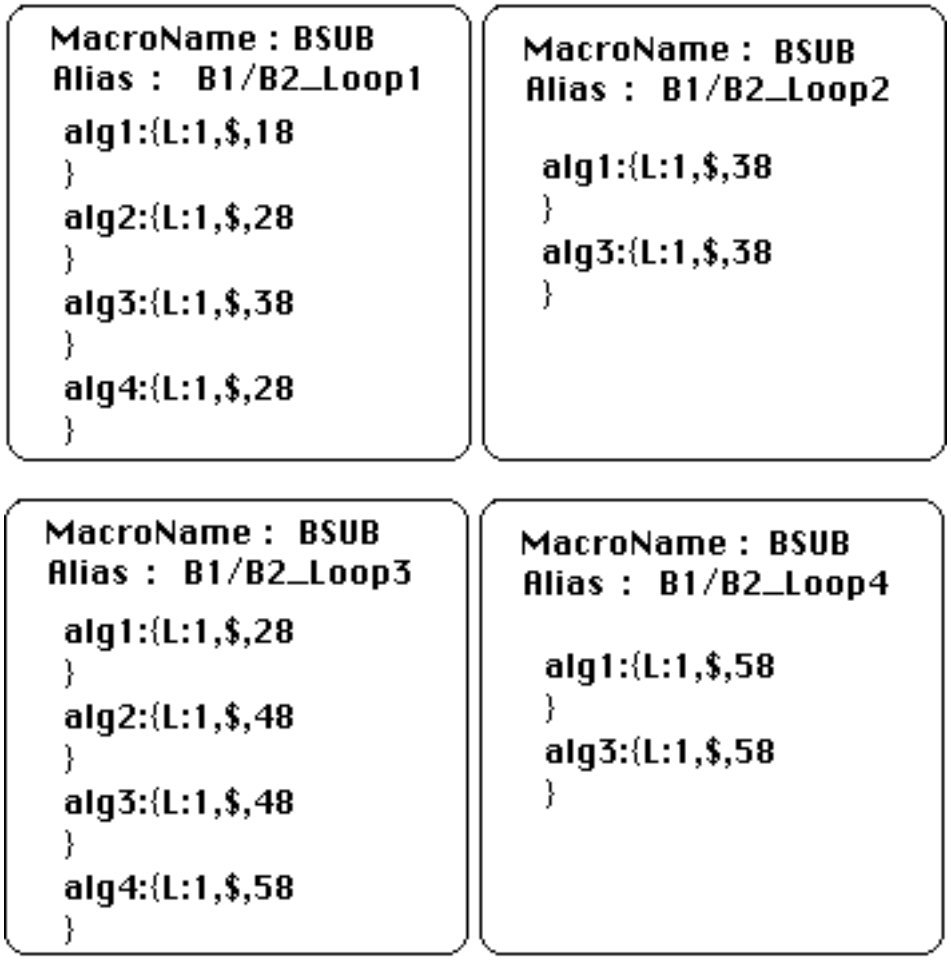


Figure 32 Invoking modifier lists associated to macro places within subnet **B1/B2_Loop**.

Places within net **B1/B2_Loop** invoke macro net **BSUB** (see Fig. 33) that is specially designed for describing the structure of melodic music objects, the **B** family; they are 18-bars objects all through the four loops of Bolero.

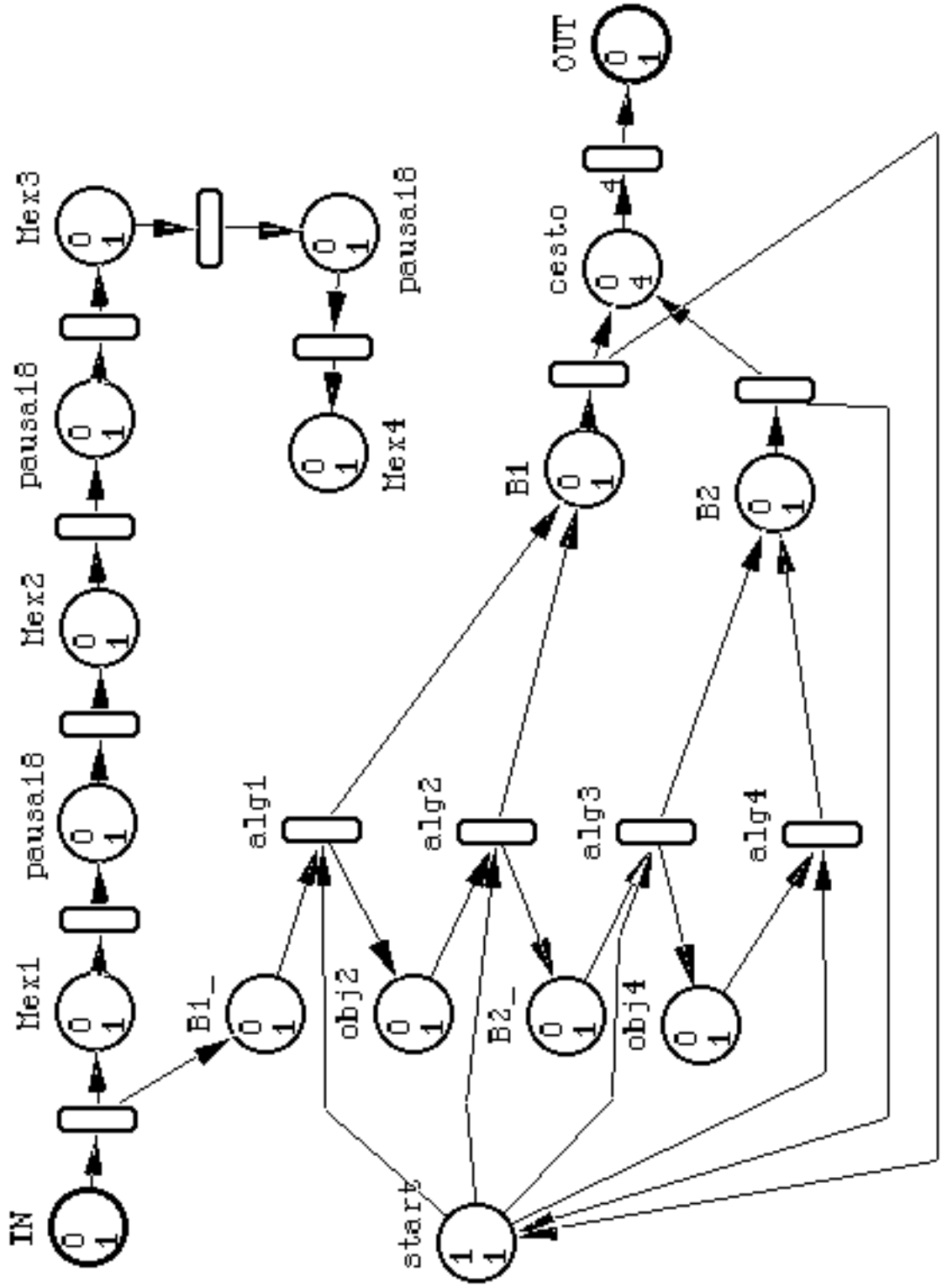


Figure 33 Macro net **BSUB**.

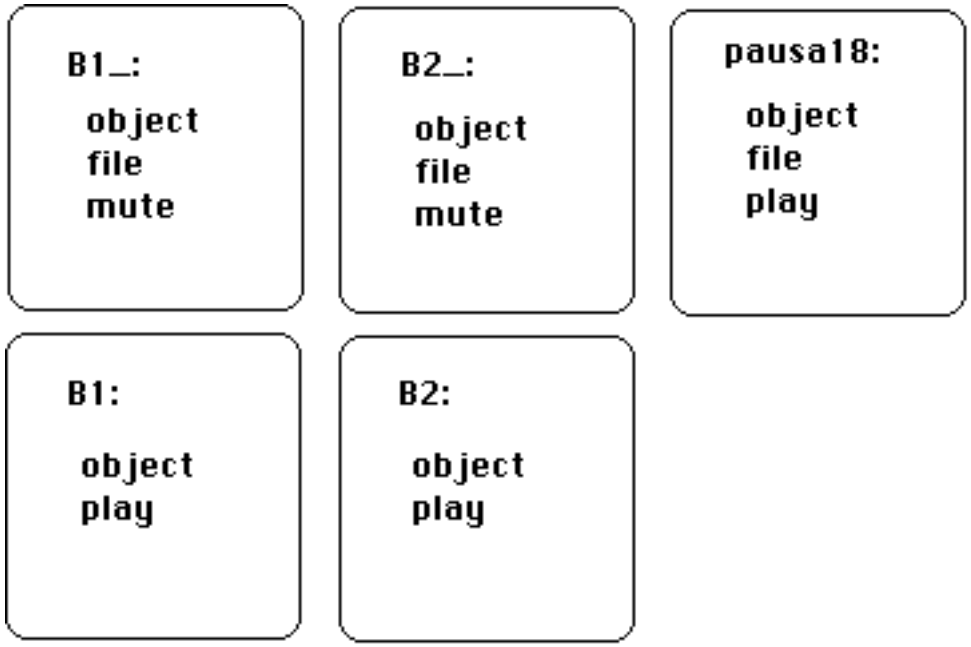
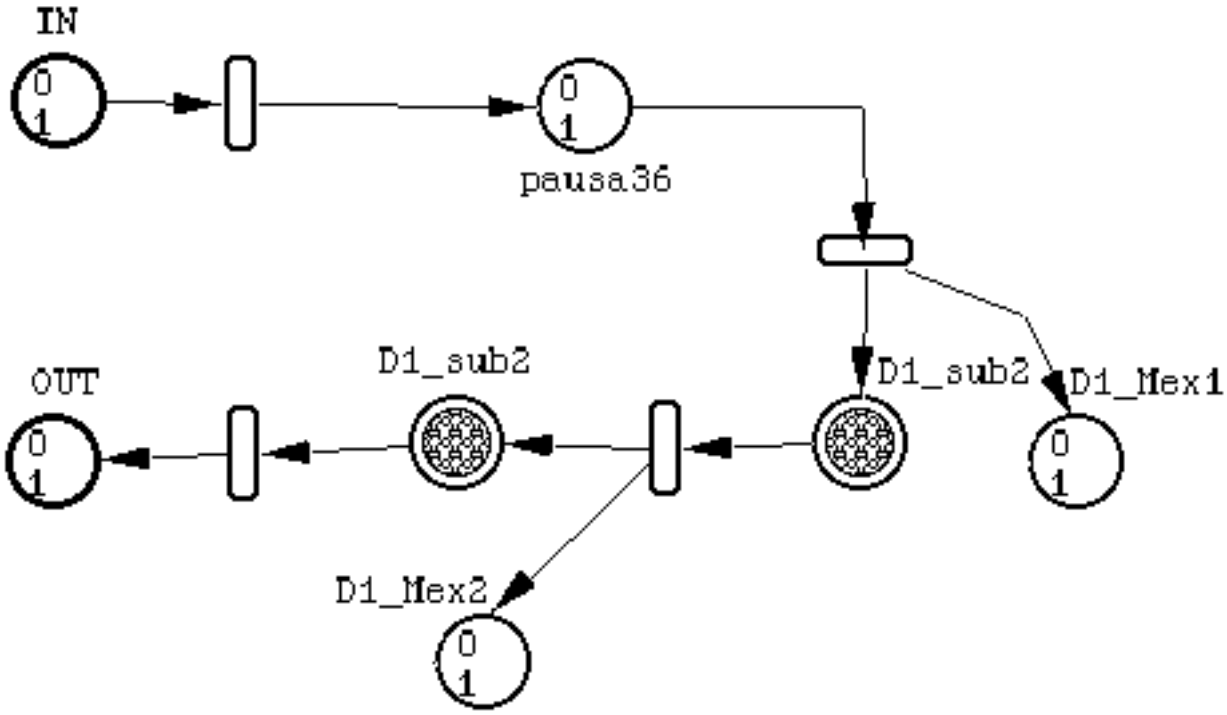


Figure 34 Attributes of places within macro subnet **BSUB**.

Figure 34 shows the attributes of places within macro subnet **BSUB**. Music objects **B1** and **B2** are associated to places **B1_** and **B2_** respectively; places **B1** and **B2** are enabled to play and have associated the music objects that are transformed by algorithms associated to transitions **alg1**, **alg2**, **alg3**, **alg4** by places of subnet **B1/B2_Loop**. As it is usual in our model, places **Mex1**, **Mex2**, **Mex3**, **Mex4** have associated commands for timbral textures' changes; they are executed at 18-bars intervals one each other.

3.4.4. Harmonic Nets

In this paragraph we speak about one-bar harmonic objects **D1** and **G**; they are harmonized underlying the execution of both major and minor melodic themes **B1** and **B2**. The following nets describe the tables we have seen in § 2.4.2.



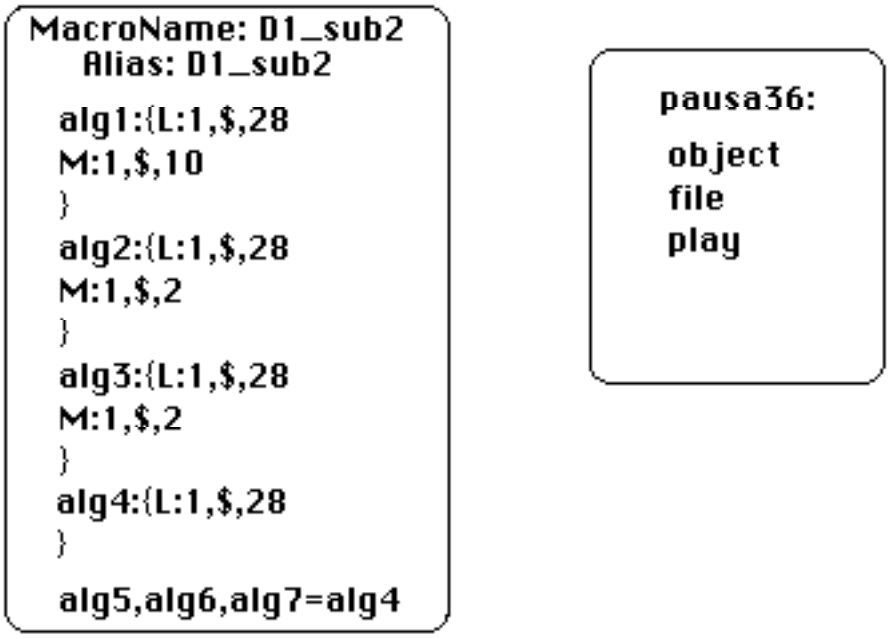
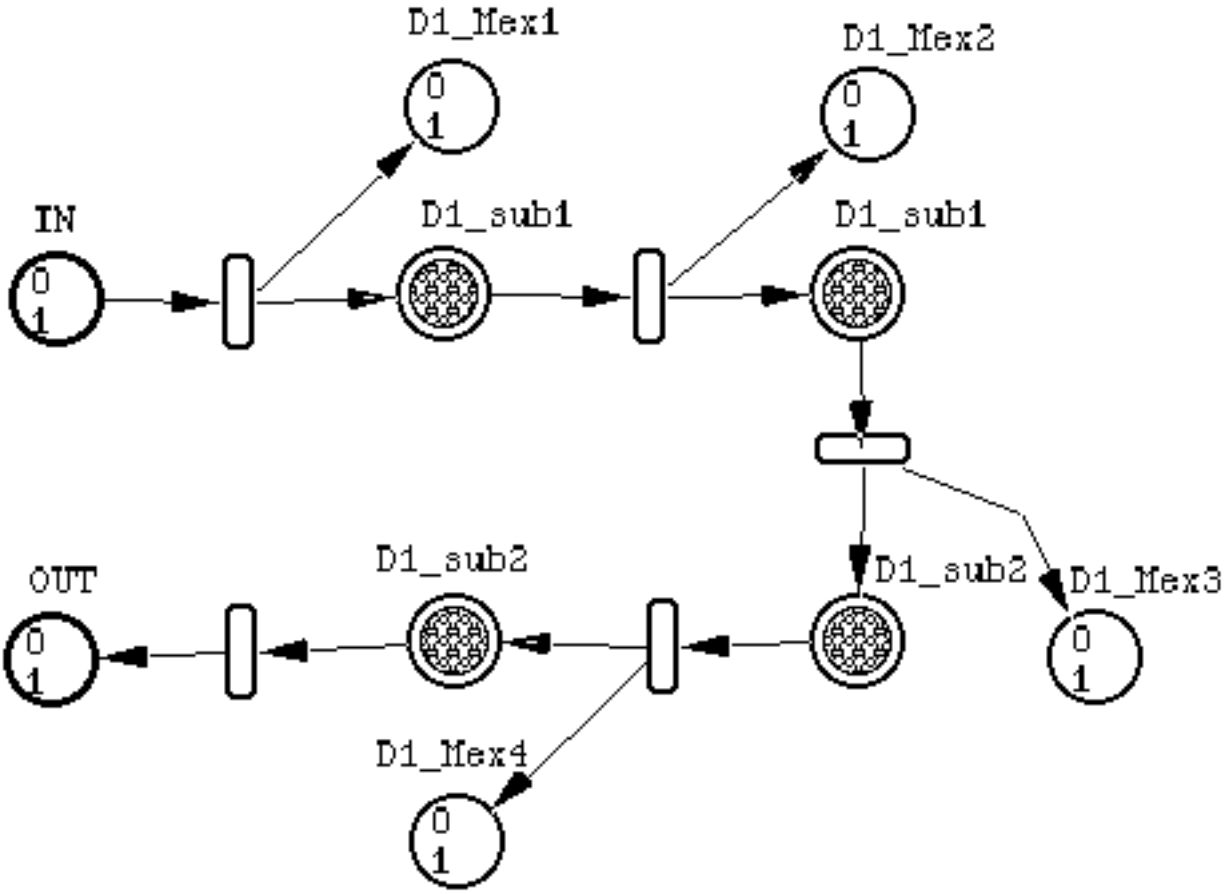


Figure 35 The subnet **D1_Loop1** associated to place **D1_Loop1** within subnet **D1_Loop** with related place attributes and invoking modifier list for subnet **D1_sub2**.

Subnet **D1_Loop1** (see Fig. 35) describes the behaviour of music object **D1** within the first loop of Bolero. Place **pausa36** has a rest associated as the music object in order to delay the appearance of **D1** in the middle of the loop. Places **D1_Mex1** and **D1_Mex2** have commands for changing timbral textures associated. **D1_sub2** is the macro net which describes the harmonizing process of music object **D1** underlying the minor theme **B2**.



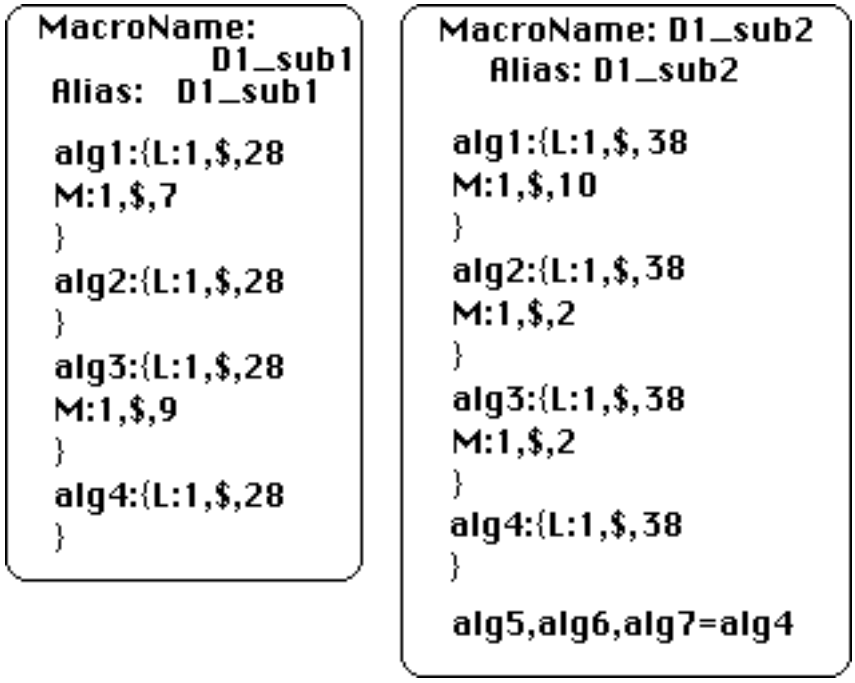


Figure 36 Subnet **D1_Loop2** with invoking modifier lists associated to places **D1_sub1** and **D1_sub2**.

Fig. 36 describes subnet **D1_Loop2** that refines the homonymous node contained within net **D1_Loop**. In this case object **D1** is also present under the major melodic theme **B1**; subnet **D1_sub1** describes the behaviour of **D1** under **B1**; in the same figure there are also the invoking modifier lists associated to places **D1_sub1** and **D1_sub2**.

Subnets **D1_Loop3** and **D1_Loop4** are not represented here because they have the same structure of subnet **D1_Loop2**. The invoking modifier lists associated to places **D1_sub1** and **D1_sub2** within subnets **D1_Loop3** and **D1_Loop4** are shown in Figg. 37 and 38 respectively.

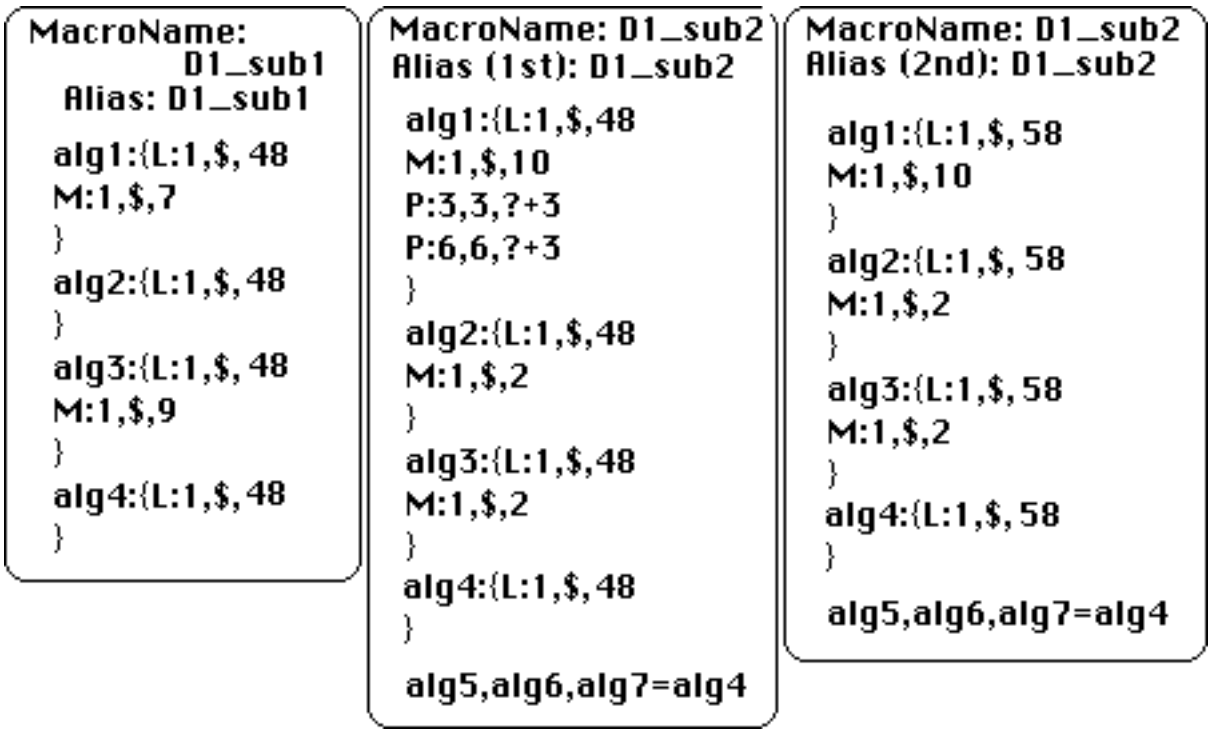


Figure 37 Invoking modifier lists associated to places **D1_sub1** and **D1_sub2** within subnet **D1_Loop3**.

As we can see in Fig. 37, different invoking modifier lists are associated to the two macro places **D1_sub2**; in fact, there is a two-bars structure which acts like a bridge underlying the first execution of **B2**, as previously seen in § 2.4.2.

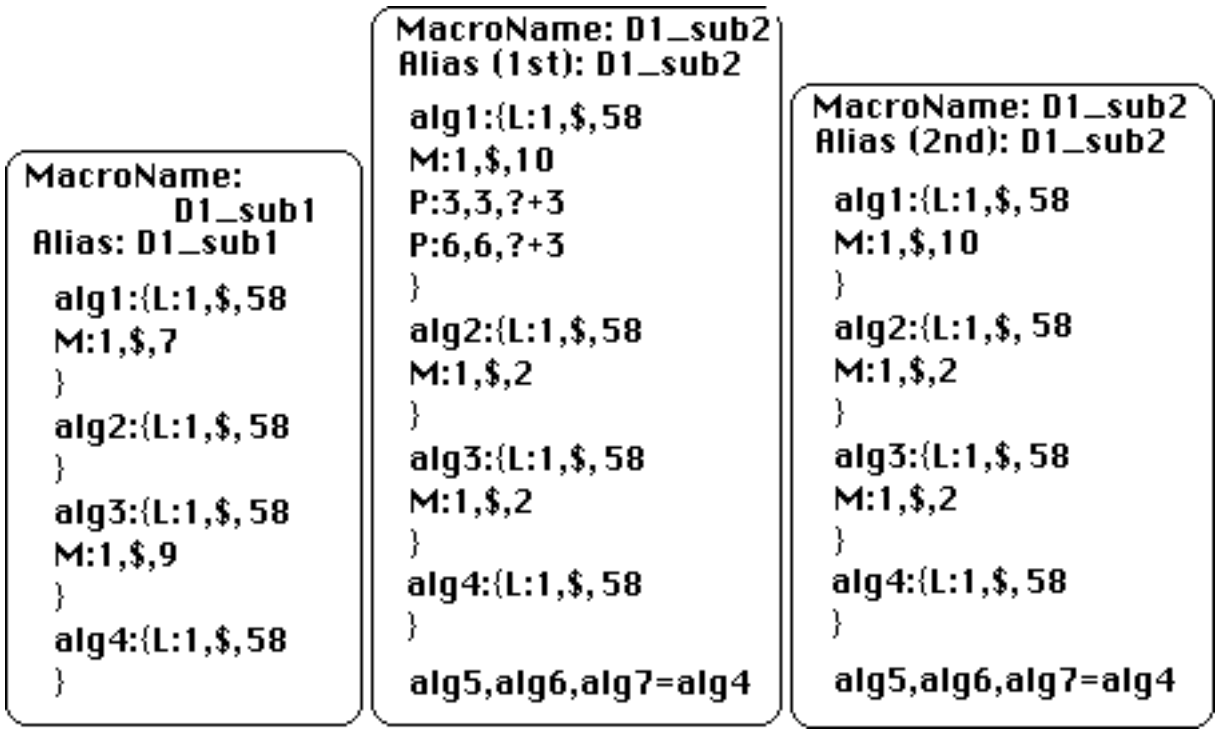


Figure 38 Invoking modifier lists associated to places **D1_sub1** and **D1_sub2** within subnet **D1_Loop4**.

Figg. 39 and 40 show macro nets **D1_sub1** and **D1_sub2** while Figg. 41 and 42 describe their place attributes respectively.

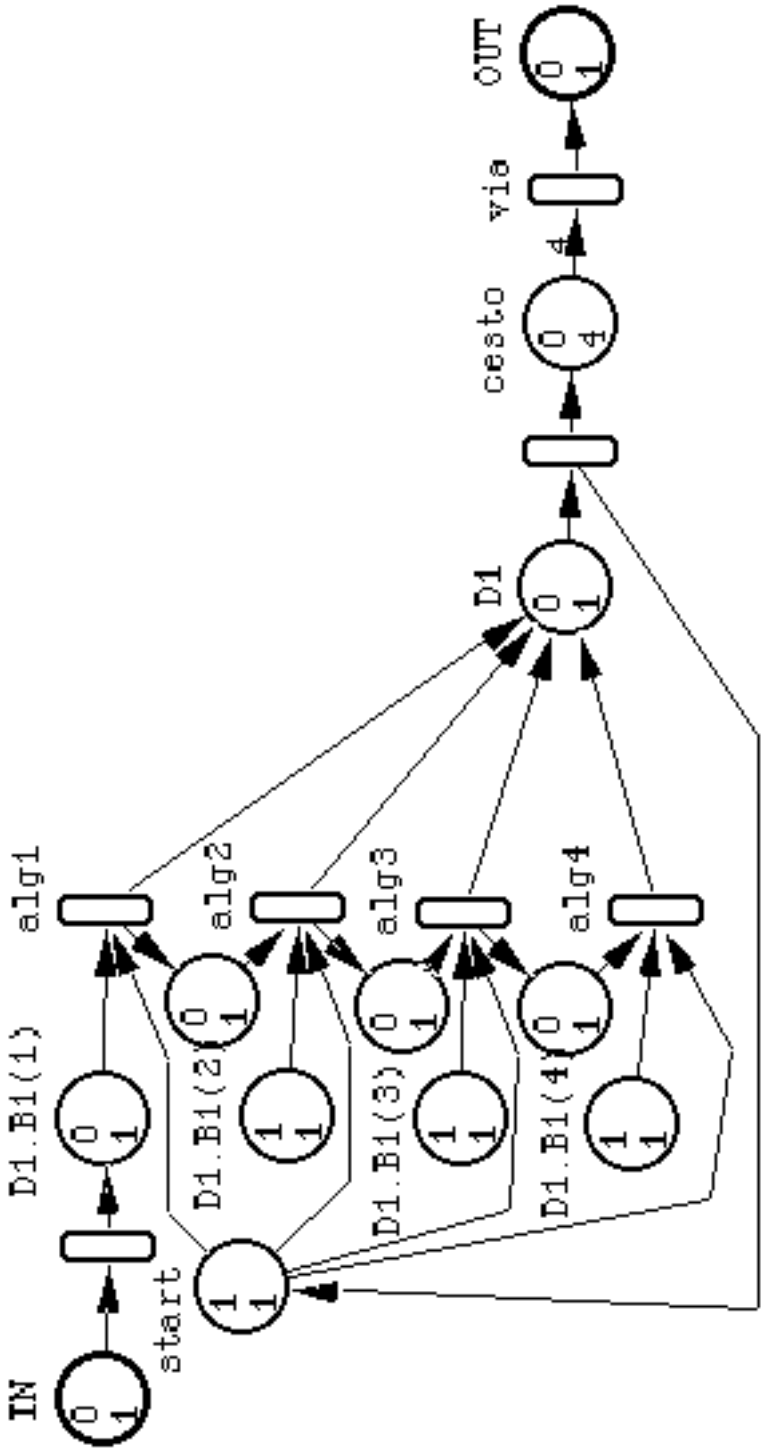


Figure 39 Macro net **D1_sub1**.

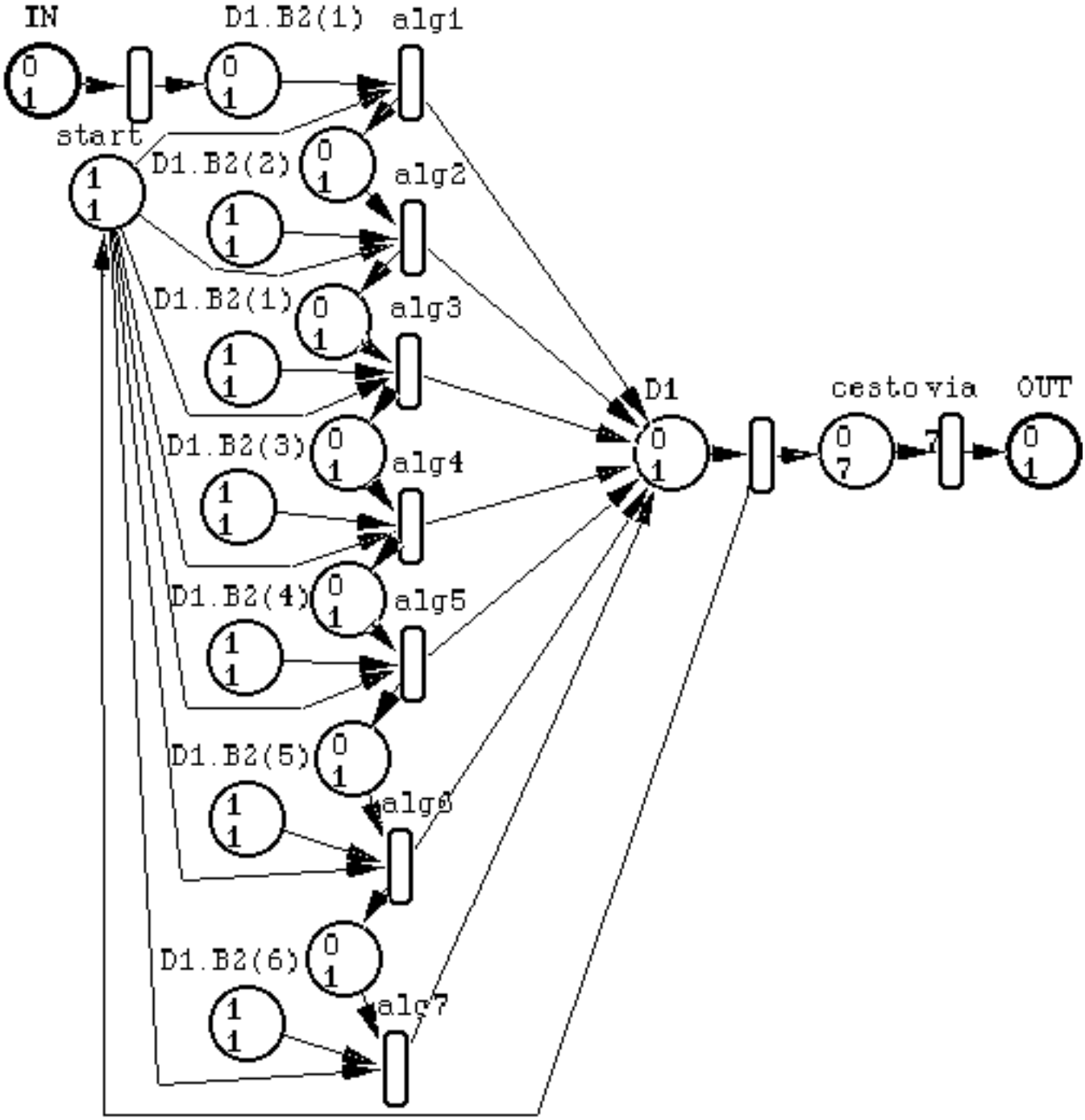


Figure 40 Macro net **D1_sub2**.

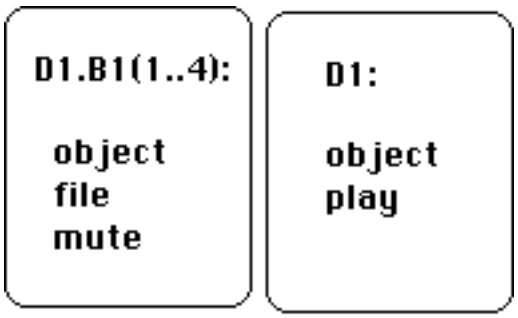


Figure 41 Place attributes within macro net **D1_sub1**.

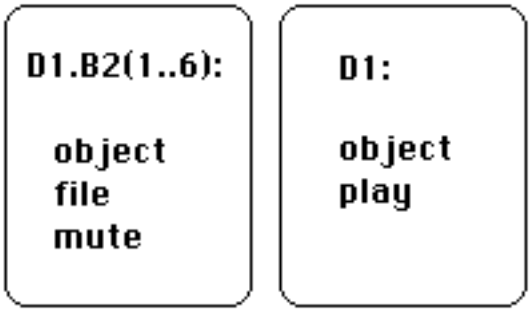


Figure 42 Place attributes within macro net **D1_sub2**.

Music object **G** has an identical behaviour with respect to **D1** throughout the four loops of Bolero, so the related nets are identical to the ones we have seen for music object **D1**.

3.4.5. The "Finale" SubModel

We can now speak about the conclusion of Bolero: the more abstract net of the Finale is shown in Fig. 43.

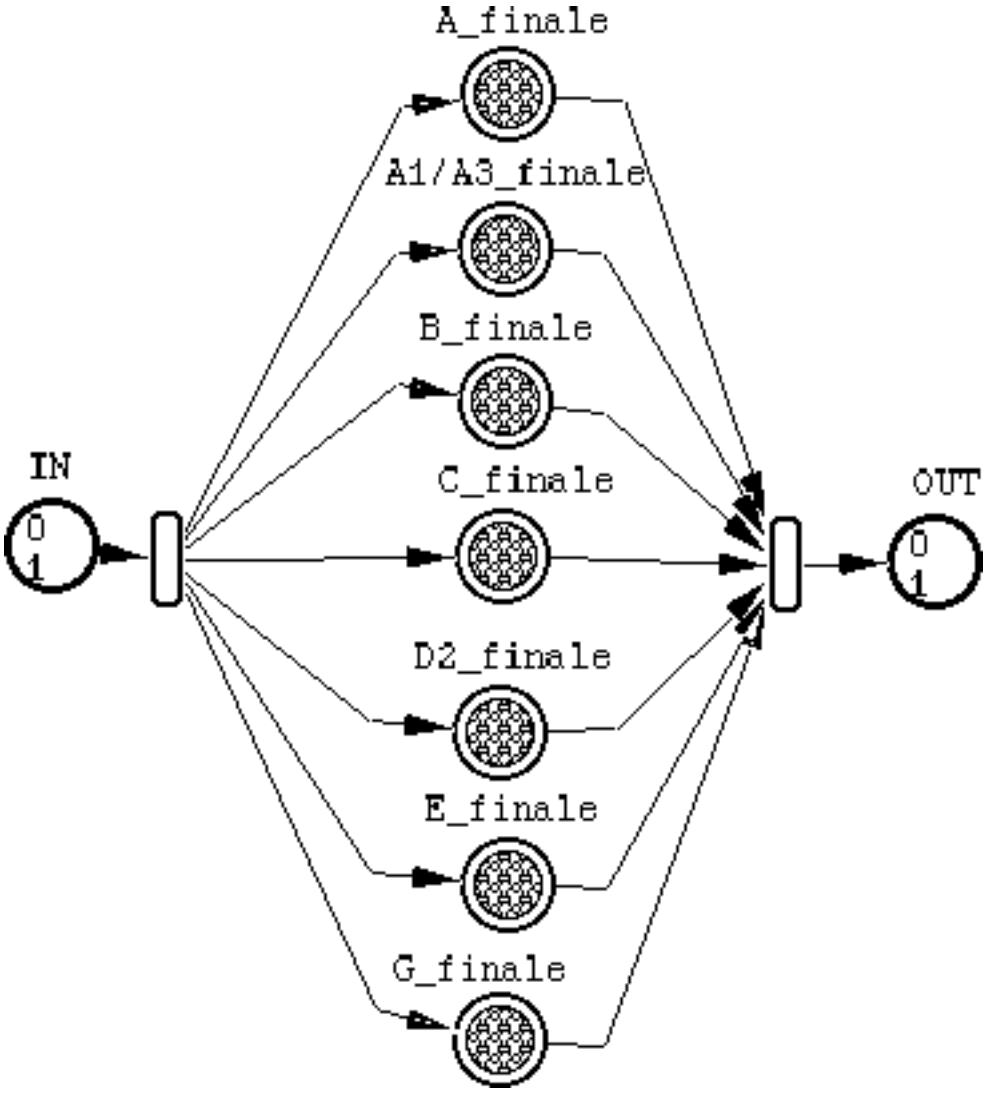


Figure 43 Subnet **Finale**.

The invoking modifier lists associated to places within subnet **Finale** are shown in Fig. 44. In this case too we have the following macro nets:
a) **FSUB**, that is used for describing the behaviour of rhythmic objects (**A1**, **A3**, **C**, **E**);

b) **B_finale**, that is used for describing the behaviour of melodic objects (**B1**, **B3**, **B4**, **Bres**).

The description of harmonic objects is made by some other simple nets.

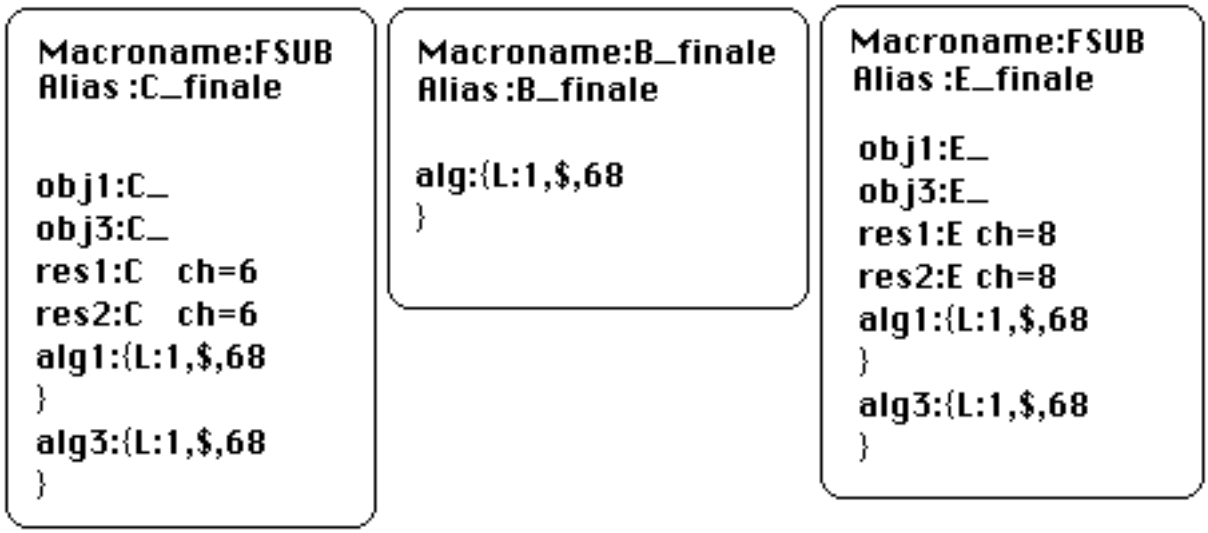


Figure 44 Invoking modifier lists associated to places within subnet **Finale**.

Fig. 45 shows macro subnet **FSUB** where the music object associate to place **res1** is played 18 times and the one associated to place **res2** is played 7 times; music objects produced by subnet **FSUB** terminate two bars before the end of the Bolero, just at the starting of the conclusive **Ares** music object (see below). Attributes associated to places within **FSUB** are shown in Fig. 46.

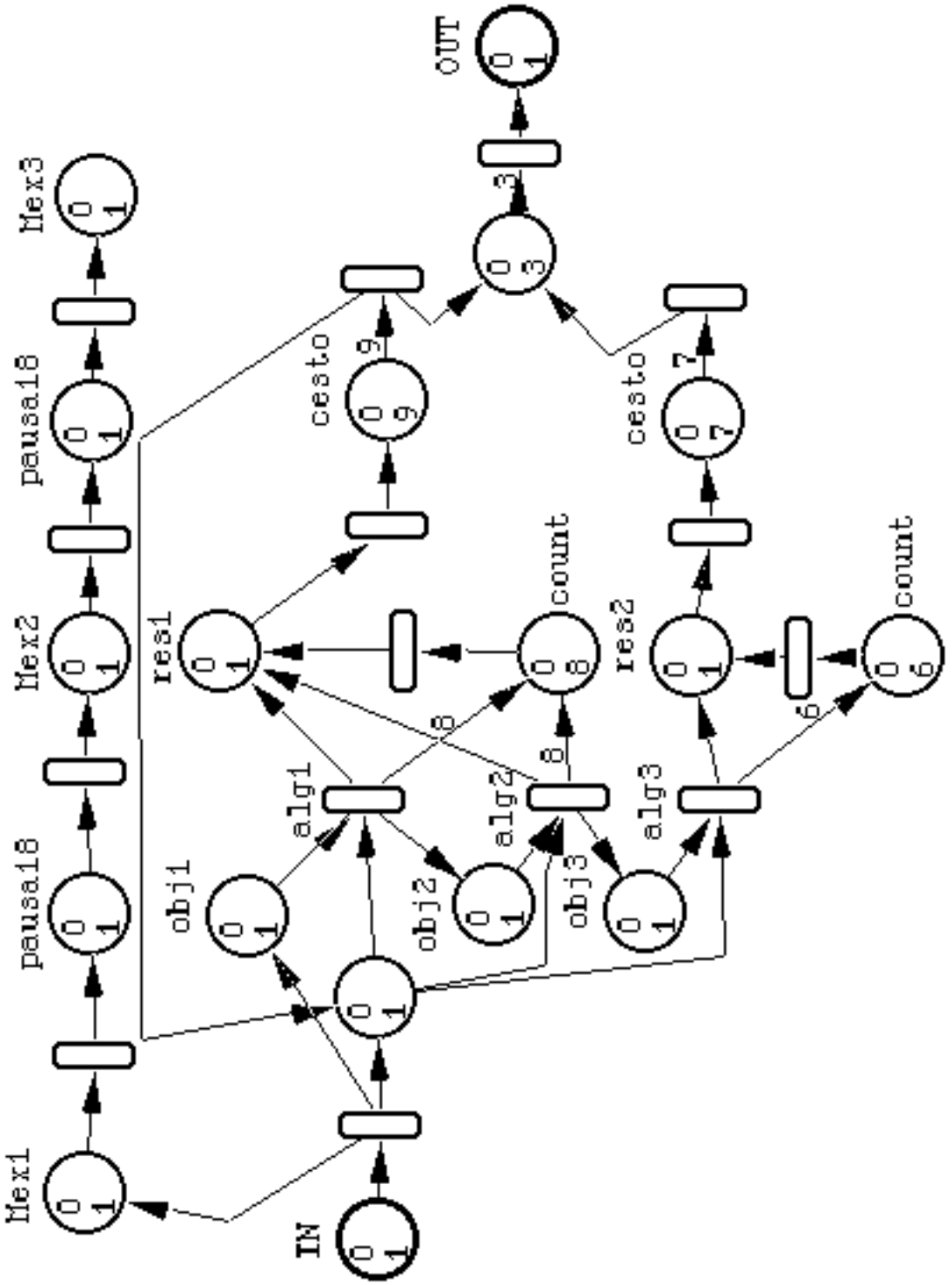


Figure 45 Macro subnet **FSUB**.

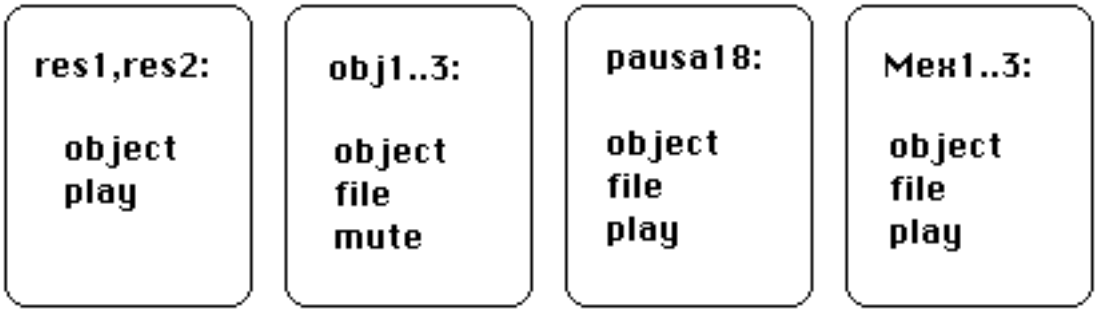


Figure 46 Place attributes within macro subnet **FSUB**.

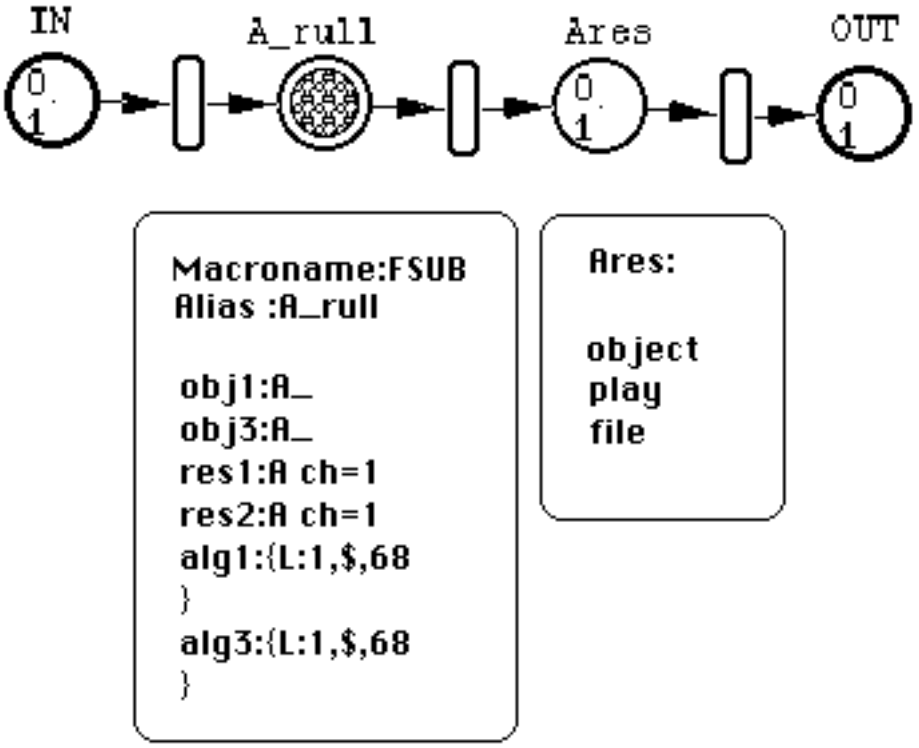


Figure 47 Subnet **A_finale** with related invoking modifier list and place attributes.

In subnet **A_finale**, there is the two-bars music object **Ares**, after the execution of macro subnet **FSUB**, that closes the Finale and the whole Bolero. Place **A1/A3_finale** is refined by the subnet shown in Fig. 48; it has the role of invoking macro subnet **FSUB** both for music object **A1** and **A3**.

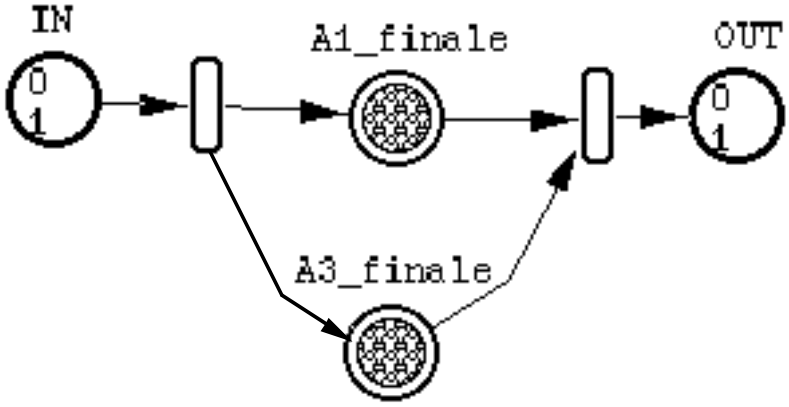


Figure 48 Subnet **A1/A3_finale**.

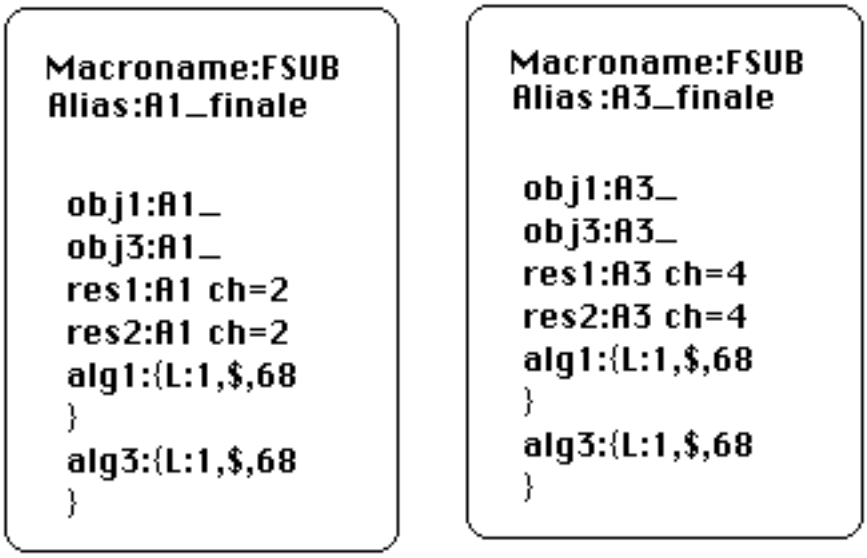


Figure 49 Invoking modifier lists associated to places within subnet **A1/A3_finale**.

Subnet **B_finale** is shown in Fig. 50 while its place attributes are shown in Fig. 51. **pausa18** and **Mex1**, **Mex2**, **Mex3**, **Mex4**, as it usual in our model, concern the management of delays and changes of timbral textures; **B1_** has music object **B1** associated before of its transformation by the algorithm associated to transition **alg**; places **B3**, **B4**, **Bres** are directly enabled to play music objects associated they have and they are not transformed by any algorithm.

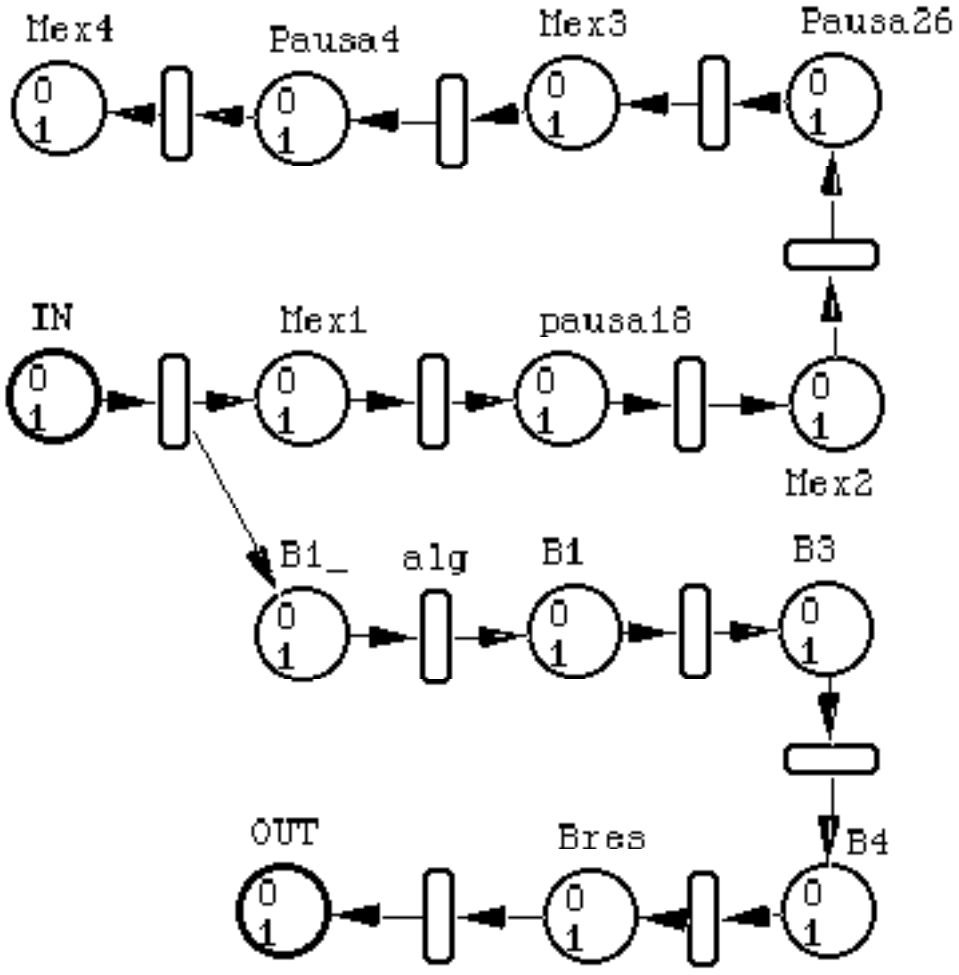


Figure 50 Subnet **B_finale**.

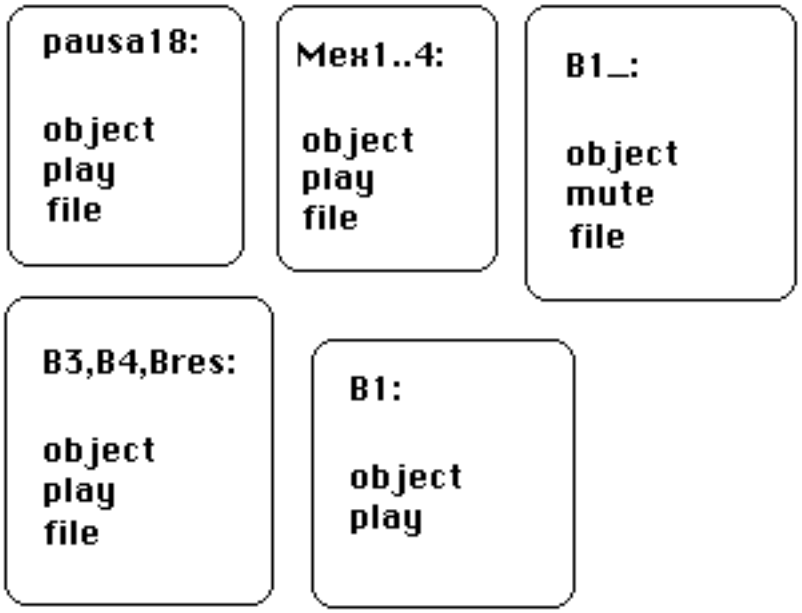


Figure 51 Place attributes within subnet **B_finale**.

Now we can see the subnets describing music objects **D2** and **G** which are harmonized in the Finale. It is suggested to read very carefully the ending part of § 2.4.2. for better understanding of the following subnets.

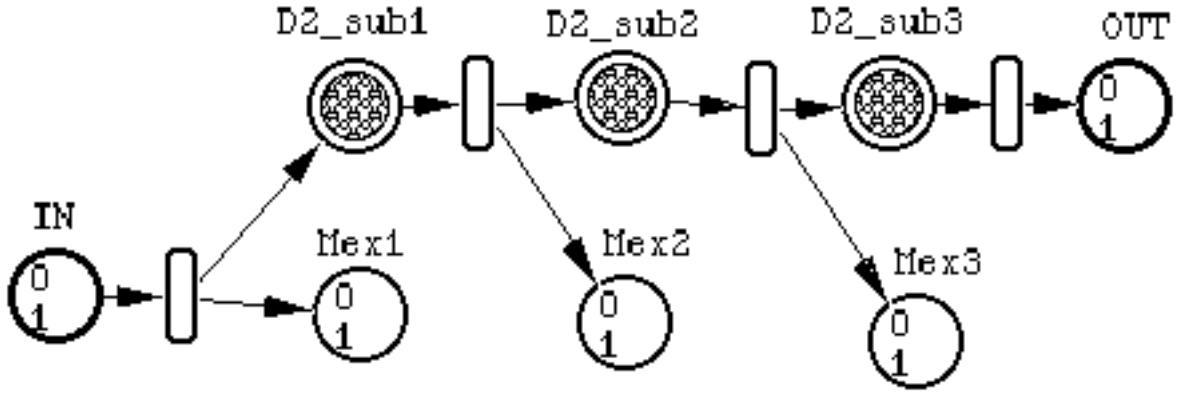


Figure 52 Subnet **D2_finale**.

Fig. 52 describes the behaviour of music object **D2**. Places **Mex1**, **Mex2** and **Mex3** have associated objects to change timbral textures. Places **D2_sub1**, **D2_sub2**, **D2_sub3** are refined by means of the subnets shown in Figg. 53, 54 and 55 respectively. Their place and transition attributes are shown in Figg. 56, 57 and 58 respectively.

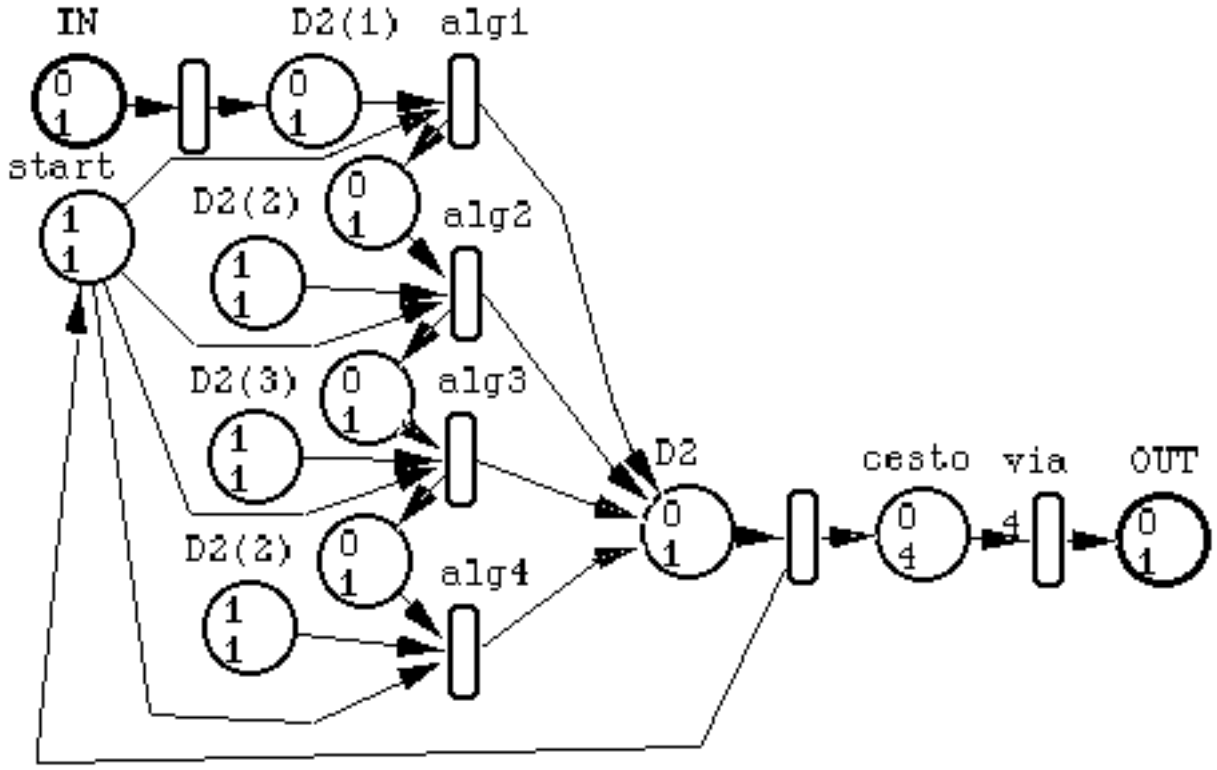


Figure 53 Subnet **D2_sub1**.

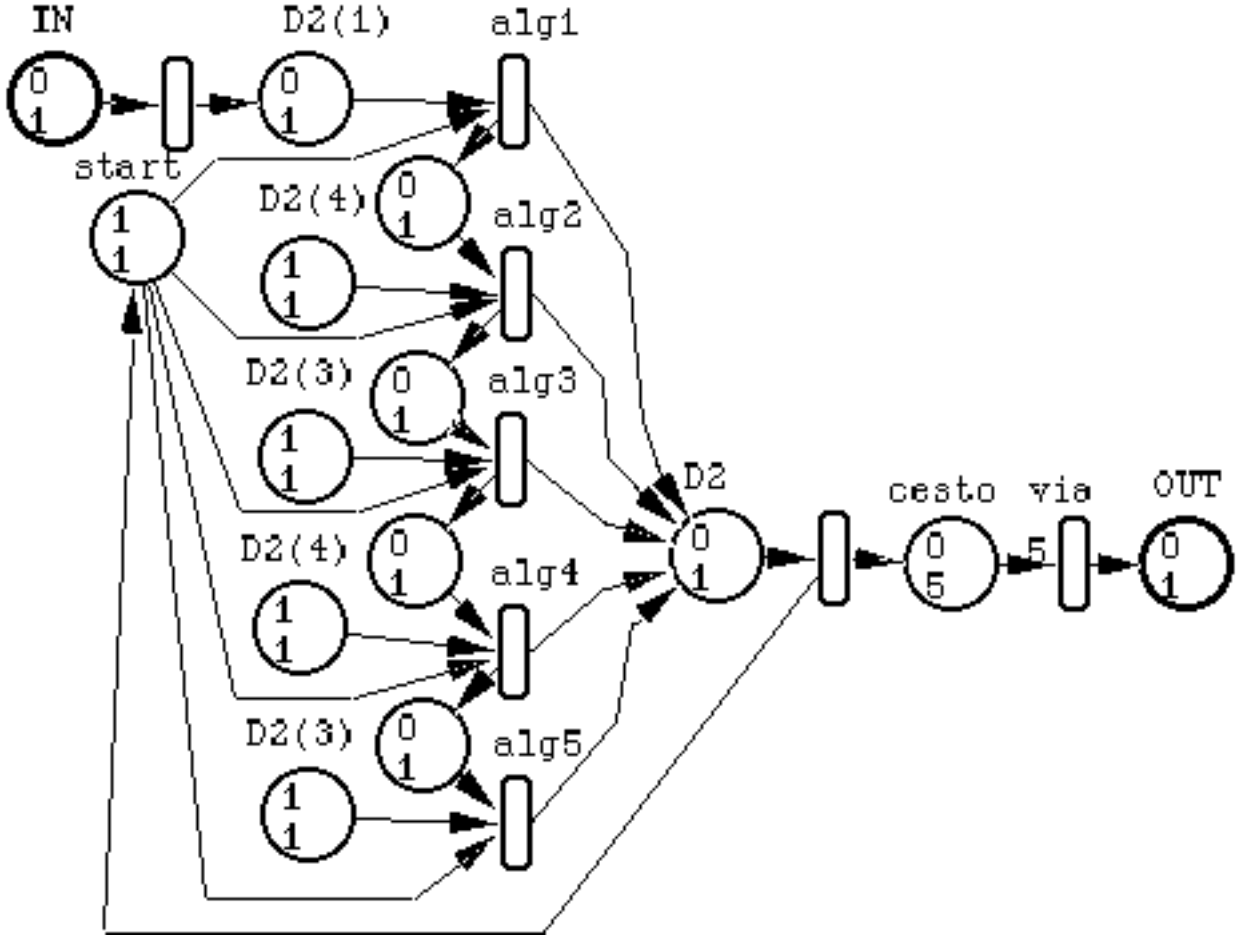


Figure 54 Subnet **D2_sub2**.

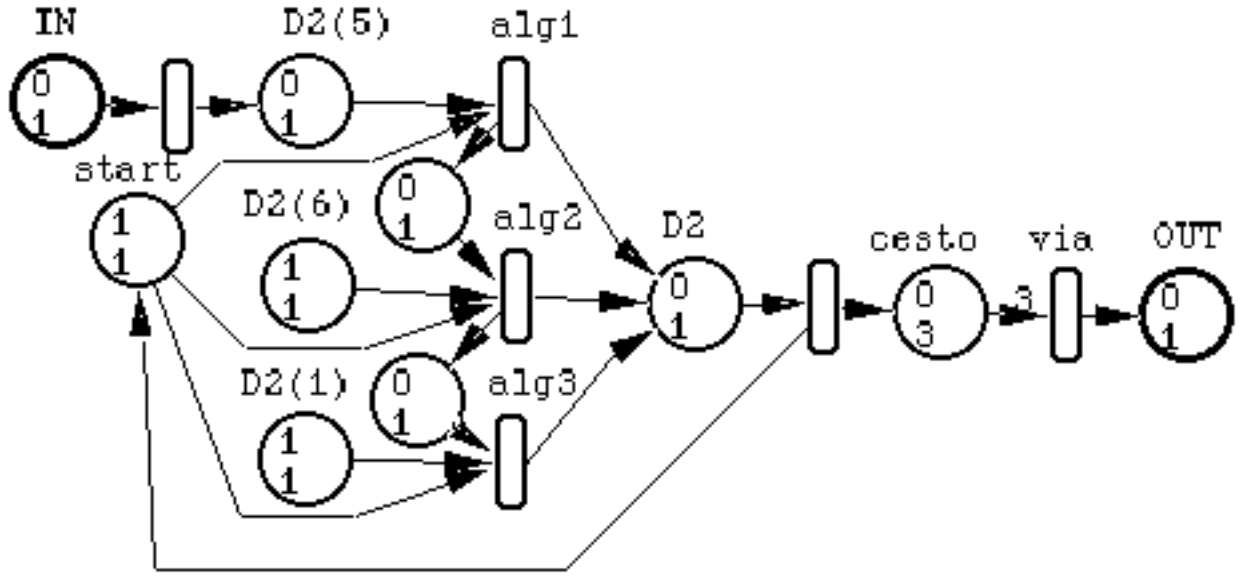


Figure 55 Subnet **D2_sub3**.

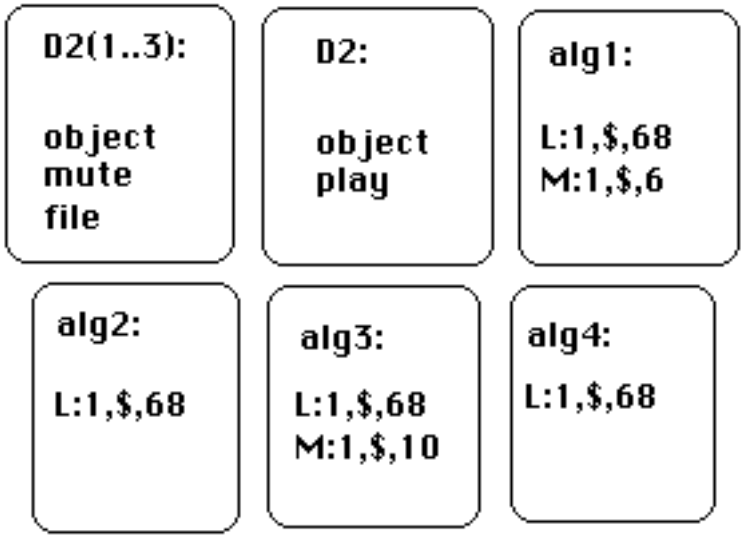


Figure 56 Place and transition attributes within subnet **D2_sub1**.

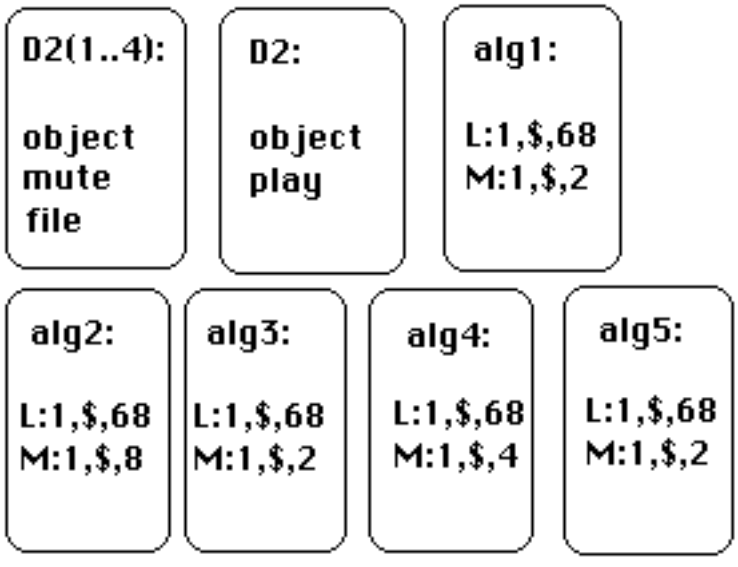


Figure 57 Place and transition attributes within subnet **D2_sub2**.

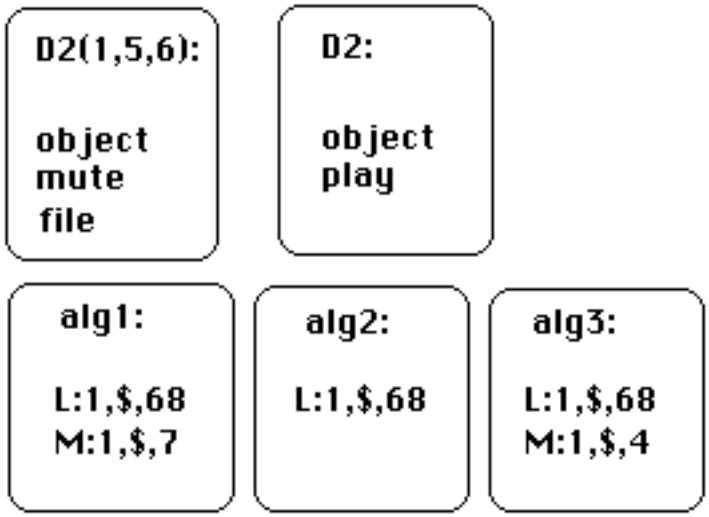


Figure 58 Place and transition attributes within subnet **D2_sub3**.

Fig. 59 describes the behaviour of music object **G** throughout the Finale phase of Bolero. Places **Mex1**, **Mex2** and **Mex3** have associated objects to change timbral textures. Places **G_sub1**, **G_sub2**, **G_sub3** are refined by means of the subnets shown in Figg. 60, 61 and 62 respectively. Their place and transition attributes are shown in Figg. 63, 64 and 65 respectively.

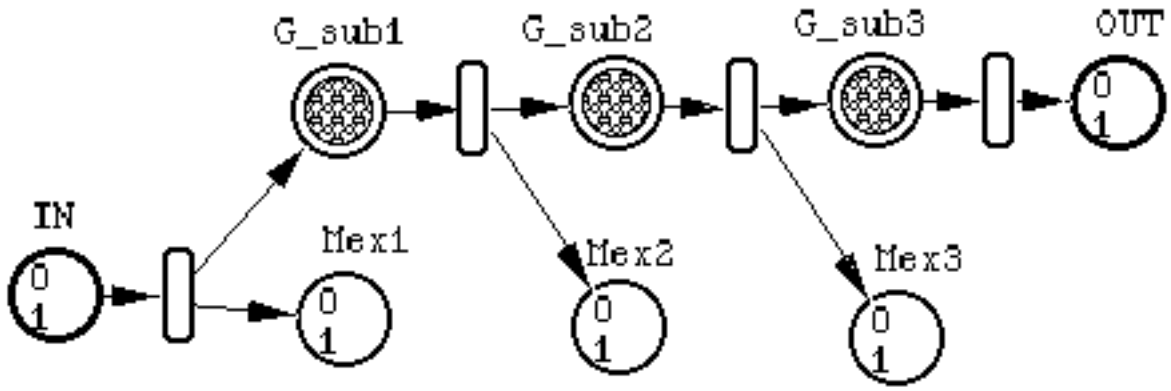


Figure 59 Subnet **G_finale**.

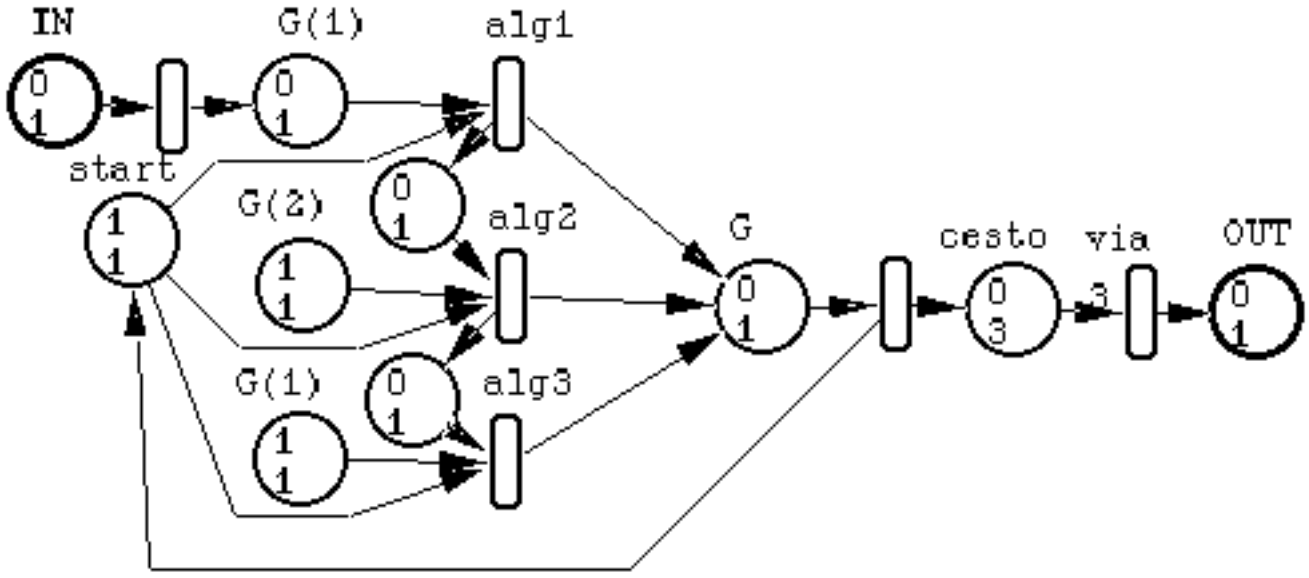


Figure 60 Subnet **G_sub1**.

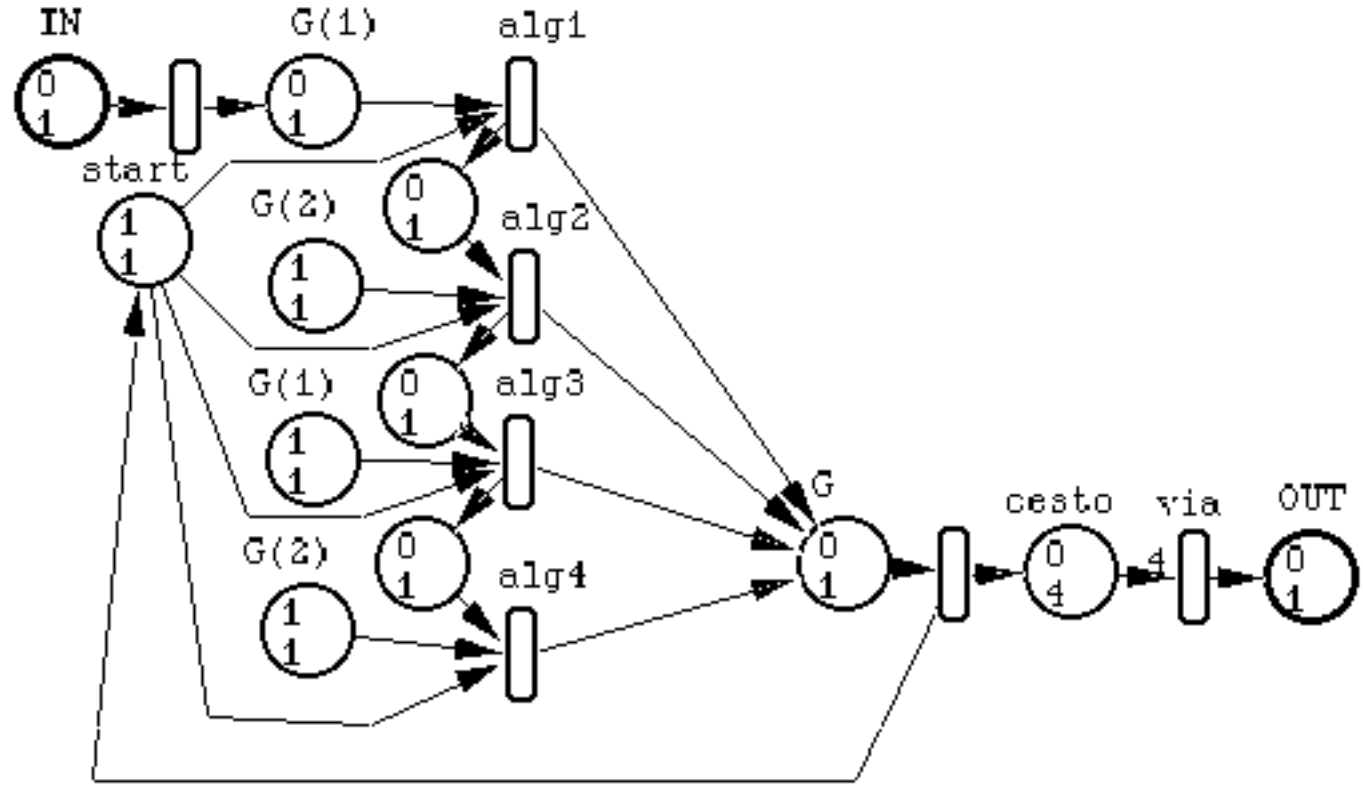


Figure 61 Subnet **G_sub2**.

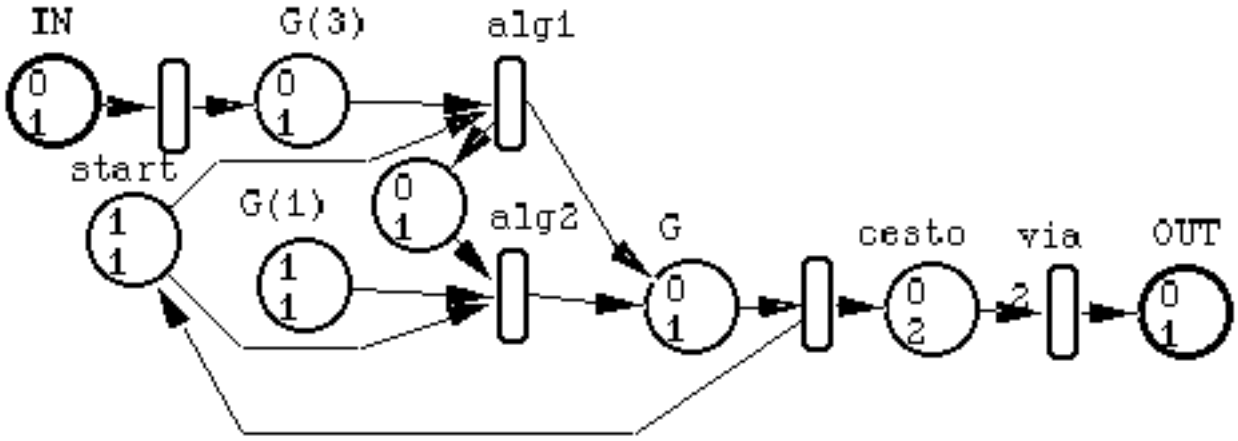


Figure 62 Subnet **G_sub3**.

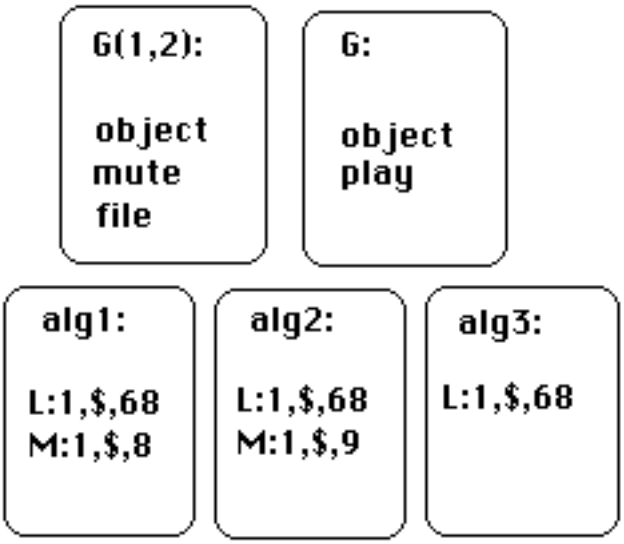


Figure 63 Place and transition attributes within subnet **G_sub1**.

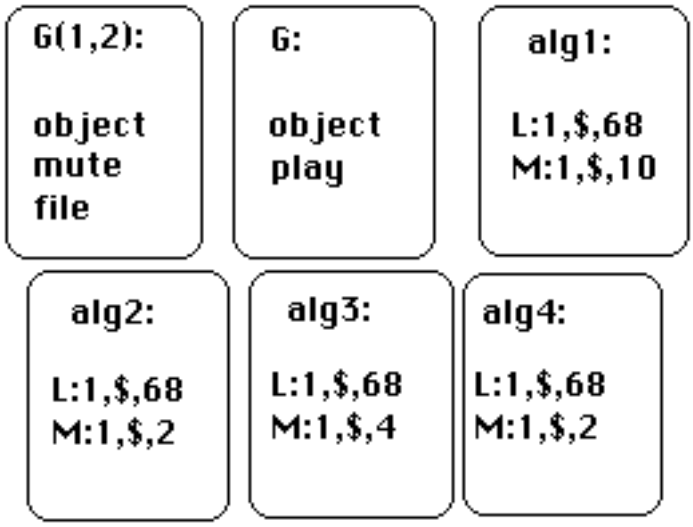


Figure 64 Place and transition attributes within subnet **G_sub2**.

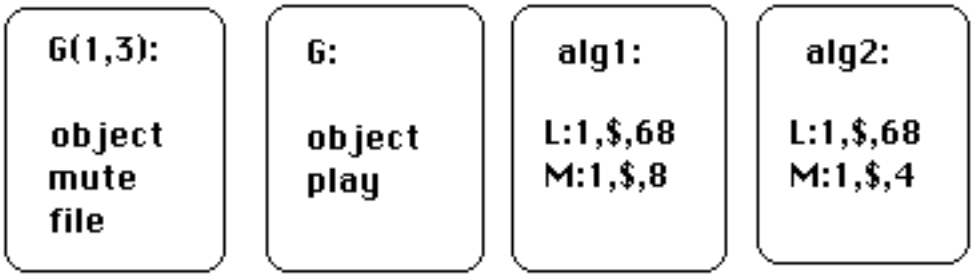


Figure 65 Place and transition attributes within subnet **G_sub3**.

4. Discussion of the Model

Petri nets in general, and particularly the implementation we have realized in the SCORESYNTH program (the PNs editor/executer we have used to develop the model of Bolero), are a highly versatile formal tool, so that very different models may be defined to describe the same music structure.

For describing Ravel's Bolero, for example, we have made a certain number of formalizing attempts to reach the definition of the model described in § 3.4. Let us consider some representative cases to allow a deeper understanding of what we are saying.

For example, we can think at the structure that controls the behaviour of music object **A** within the four loops of the piece: object **A** is played by a drum, so the transformations applied to that object concern dynamics. In the beginning we have used four subnets as it is shown in Fig. 66 where macro places invoke subnet **Loop** (see Fig. 67). The invoking modifier lists are all of the same kind; as an example we can see that of macro place **18App**:

```
Object: A_
Algorithm: {L:1,$,18
}
Counter: M=17
Basket: C=18
Basket->Way=18
```

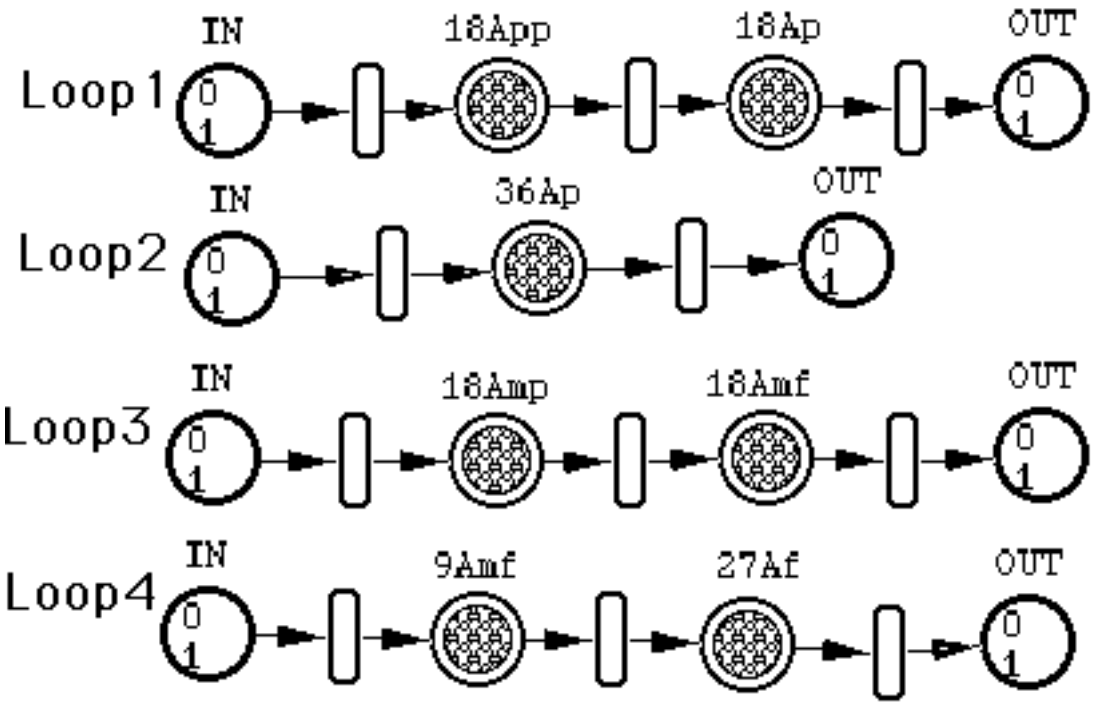


Figure 66 Alternative subnets for the description of the behaviour of music object **A**.

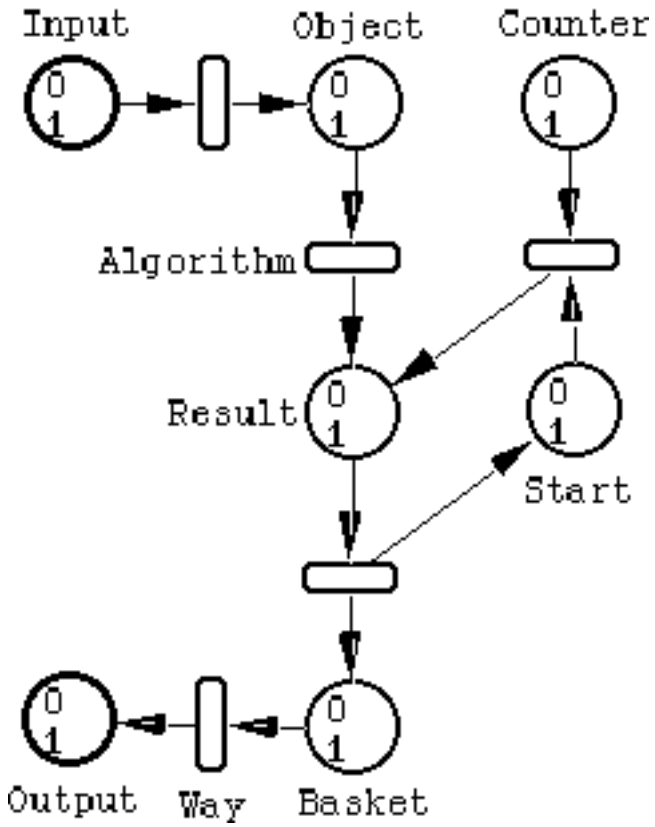


Figure 67 Macro net **Loop**.

Therefore, as well as each of them assign a music object to place **Object** and an algorithm to transition **Algorithm**, the invoking modifier lists include parameters to set markings, capacities and multiplicities so that music objects are played as many times as it is deterministically described. The execution of the nets in Fig. 66 produces the same score of the subnet **A_Loop** (see Fig. 23 above) with invoked subnet **SUB** as the output, but in the modelling approach we have previously seen the subnet **SUB** has within its own net structure the information concerning *when* transformations are to be applied to music objects; in fact, the

two-bars object that is associated to place **res** is played 9 times between any couple of transformations (that is a quarter of a loop).

We can observe that the same reasoning may be repeated for every rhythmic object; so, in our model, we have simplified high level nets (**A_Loop**, **C_Loop**, etc.) by means of a macro subnet, **SUB**, which executes all the four loops.

Another problem we have had concerns duplications within the Bolero score. In fact, before introducing the approach based on changes of timbral textures controlled by MIDI exclusive commands, we had to build very complex nets when a music object was played by many instruments.

For example, we show how nets describing melodic objects **B1** and **B2** were within the fourth loop. Fig. 68 is an alternative net to represent B1/B2_Loop, which has the same net structure of A_Loop, and related subnets (macro net BSUB with its four invoking; see Figg. 33 and 32 above). Observe that places **terza**, **quinta** and **ottava** are not directly connected by arcs to the second transition within the sequential net: it means that the synchronization process among the sequence of objects within the output score is not directly controlled by the net structure but it is implicitly described depending on the duration of music objects associated to places; in other words, if all the music objects associated to places **tonalità**, **terza**, **quinta** and **ottava** have the same duration there is synchronization, otherwise there is overlapping of music objects with respect to the different execution steps of the sequential net.

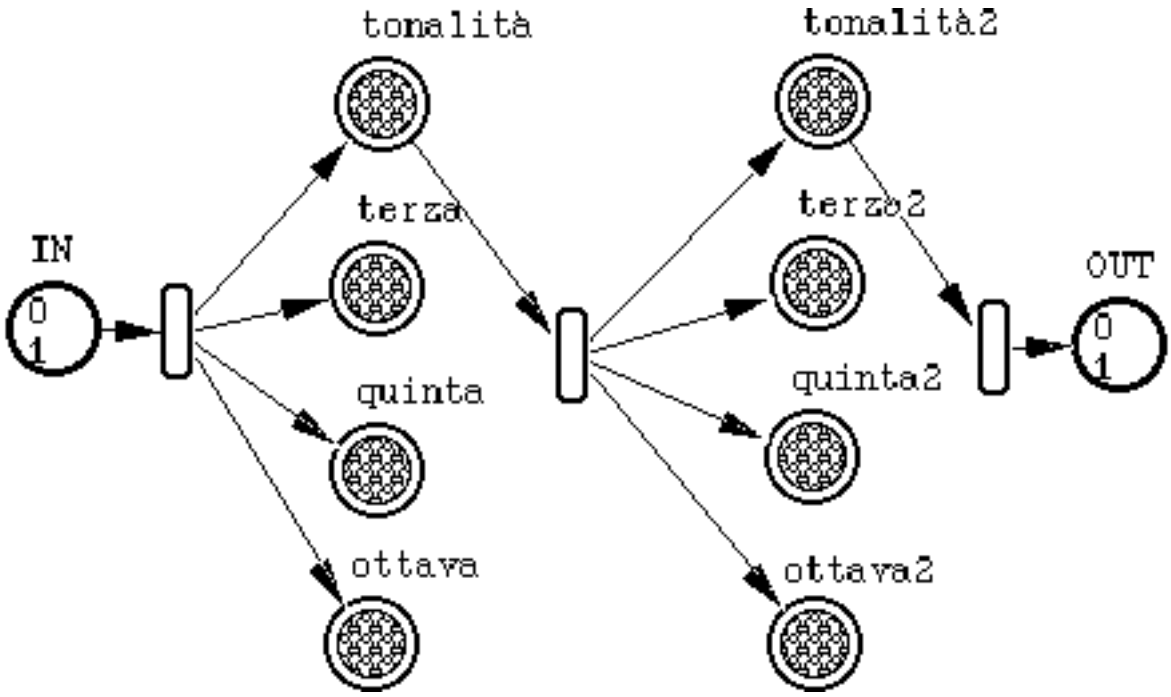


Figure 68 Alternative net to represent **B1/B2_Loop**.

Figg. 69, 70, 71 and 72 refine places **tonalità**, **terza**, **quinta** and **ottava**; the subnets which refine the other places of the net are very similar to these.

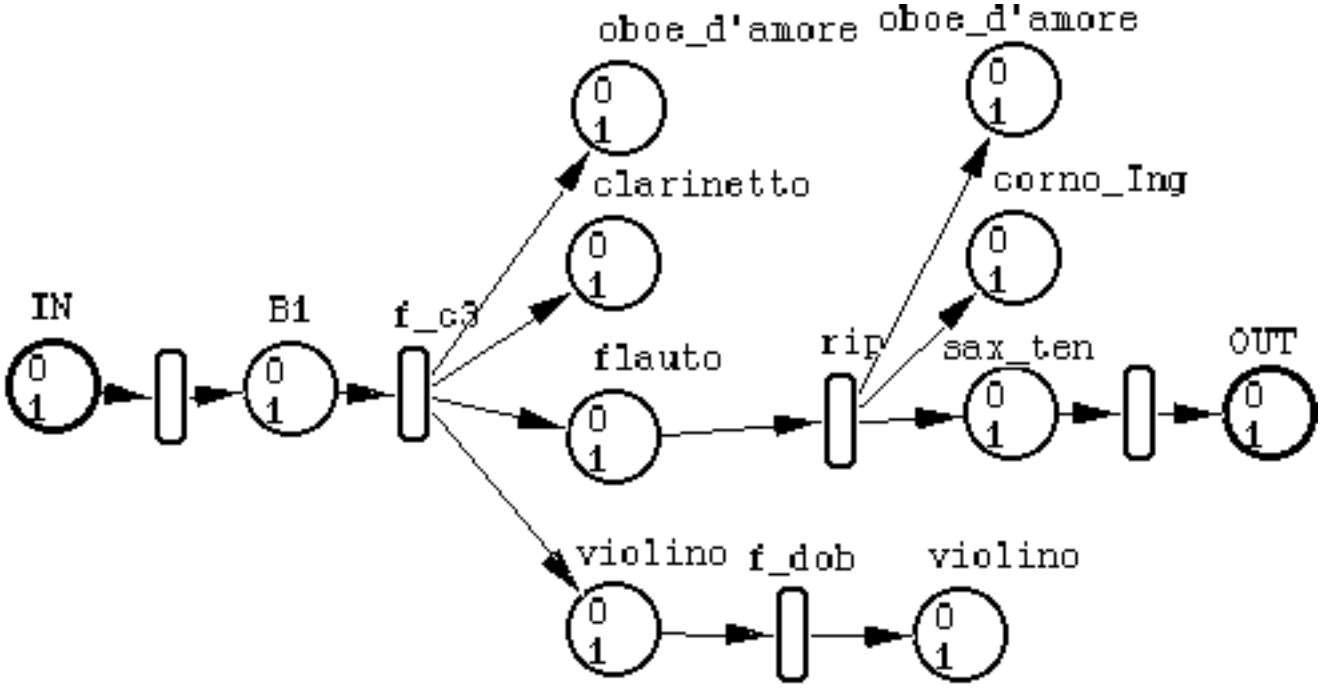


Figure 69 Subnet **tonalità**.

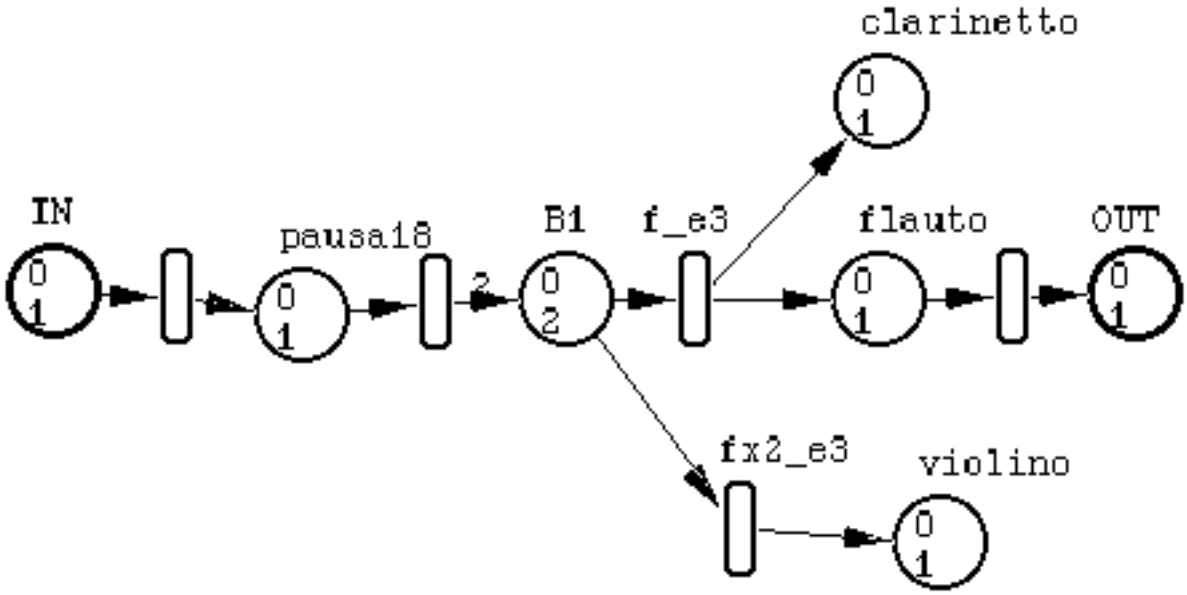


Figure 70 Subnet **terza**.

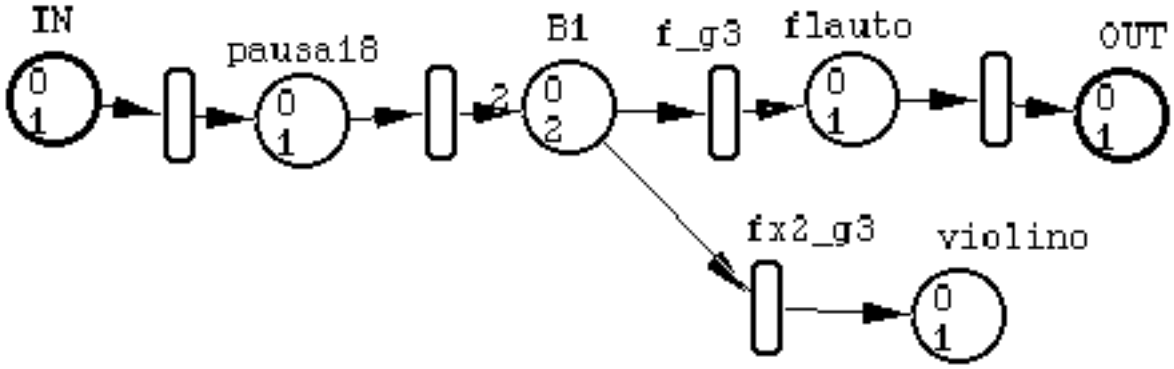


Figure 71 Subnet **quinta**.

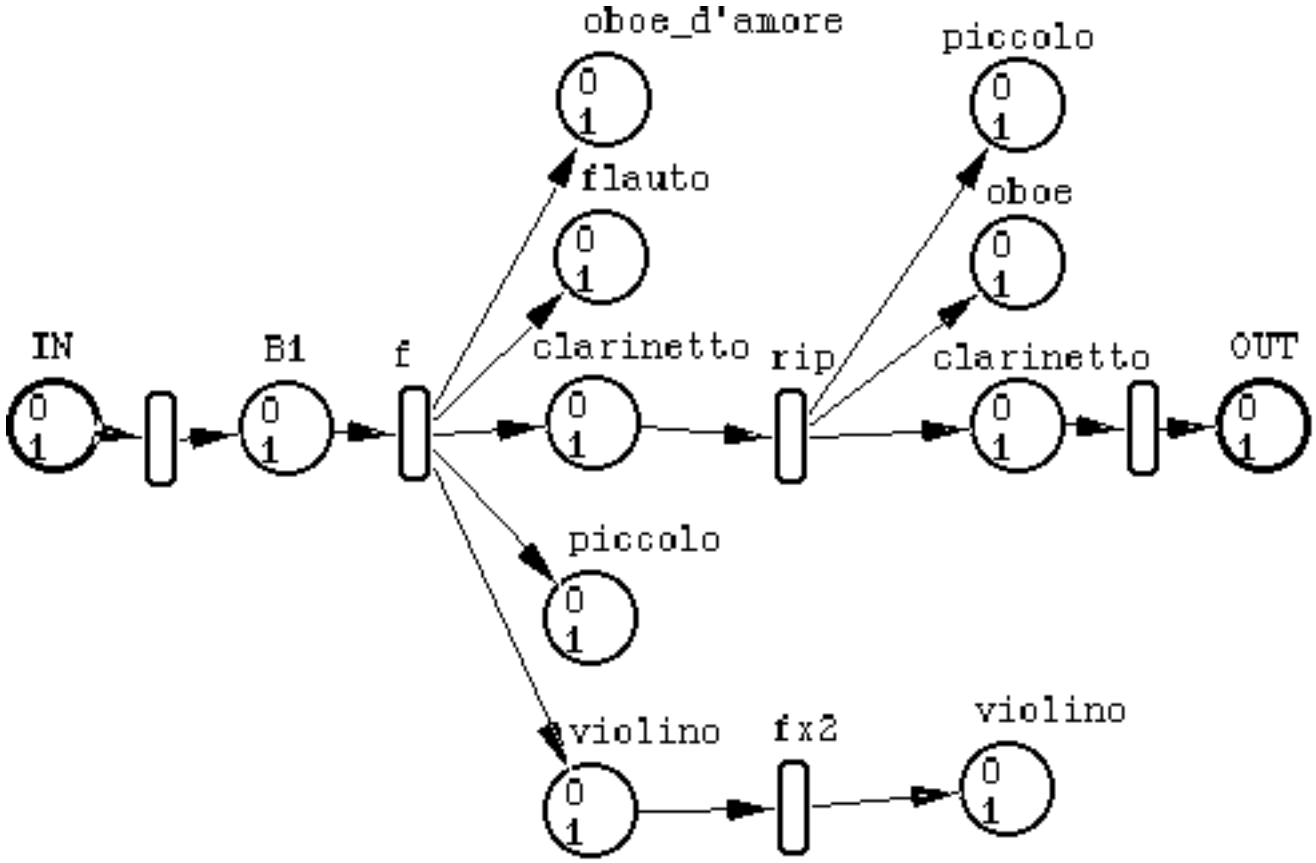


Figure 72 Subnet **ottava**.

If one looks at the model we have defined, he can see that, due to the approach based on the changes of timbral textures, all duplication substructures within the kind of nets in Figg. 69, 70, 71 and 72 can be eliminated and all the fourth loop, with respect to objects **B1** and **B2**, can be described by the modifier list associated to place **B1/B2_Loop4** invoking macro subnet **BSUB** we have seen in Fig. 32. The simplification is clear.

We can continue on discussing more solutions for the modelling of Bolero, but we stop here; it seems already clear, we think, that Petri nets are suitable for the modelling of complex music scores and that there are many different meaningful solutions for the modelling of the same piece.

Furthermore, a particularly interesting line of research lies in the possibility of transforming musical structures by modifying some characteristics within PNs models.

It is convenient to point out that deep changes within models (that is, changes within low level PNs) do not necessarily affect deep changes within music results (from the listener perceptive point of view); and viceversa. Some examples follow which show these concepts.

If we change the marking of the place **count**, the capacity of the place currently set to 4 and the multiplicity of the arc currently set to 4 within macro net **Loop1-4** (see Fig. 22 above) and we modify related subnets in a suitable way (adding or deleting net substructures within nets **A_Loop**, **A1/A2_Loop**, **B1/B2_Loop**, **C_Loop**, **D1_Loop**, **G_Loop** and **E_Loop**, depending on the number of loops we define), we bring about a very deep transformation of the Bolero score. In fact, if we put three only tokens into the counter place, the execution of the model will synthesize a shorter Bolero, made by three development cycles, reduced dynamics, reduced harmonization and reduced instrumentation. Viceversa, if we put five tokens into the counter place, the execution of the model will synthesize a longer Bolero, made by five development cycles, increased dynamics, increased harmonization and increased instrumentation.

Another example of simple alteration which brings about a deep transformation is the following: if we change the rhythmic content of the **A**-family objects, the execution of the model will synthesize a totally different score which may be, for instance, the "Waltz" or even the "Tarantella" by Ravel.

A chance is to fine adjust dynamic, harmonic and instrumental developments within Ravel's Bolero to correct the imperfections of the score execution which are due to sound synthesis devices. This kind of work produces little perceptive results, but requires a lot of experimental efforts.

At last, we mention that one can introduce non-deterministic structures within the model so that the output score is unpredictable as much as the non-deterministic structures affect the most relevant generative structures of the model.

Acknowledgements

A lot of thanks are due to the whole L.I.M.-Laboratorio di Informatica Musicale staff for the scientific and technical support provided. Special thanks are due to: A. Sametti and S. Scolaro for the development of the ScoreSynth software, to S. Scolaro again for his help in the implementation of the Petri Nets model of Bolero by the ScoreSynth software and to A. Camurri for his "historical" role in our research projects about Petri Nets & Music.

5. References

[1] M. Ravel: "*Bolero*", Durand S.A., Paris, 1929.

[2] G. Degli Antoni, G. Haus: "*Music and Causality*", Proc. of the 1982 ICMC, Venezia, pp. 279-296, CMA Publ., San Francisco, 1983.

[3] G. Degli Antoni, G. Haus: "*Netz Representationen von Musikstucken*", in "*Musik Psychologie. Ein Handbuch in Schlusselbegriffen*", Bruhn/Oerter/Rosing Ed., pp. 141-148, Urban & Schwarzenberg, Munchen, 1985.

[4] S. Pope: "*The Development of an Intelligent Composer's Assistant*", Proc. 1986 ICMC, 16 pp., Den Haag, CMA Publ., San Francisco, 1986.

[5] H. J. Genrich, K. Lautenbach: "*System Modelling with High-Level Petri Nets*", Theoretical Computer Science, Vol. 13, pp. 109-136, 1981.

[6] C. A. Petri: "*Communication mit Automaten*", Schriften des Institutes fur Instrumentelle Mathematik, Bonn, 1962.

[7] C. A. Petri: "*General Net Theory*", Proc. Joint IBM & Newcastle upon Tyne Seminar on Computer Systems Design, 1976.

[8] G. Scheschonk: "*Eine einfuhrende Zusammenfassung der Petri Netz Theorie*", Universitatsbibliothek der Technischen Universitat, Berlin, 1977.

[9] J. L. Peterson: "*Petri Net Theory and the Modeling of Systems*", Prentice Hall, New Jersey, 1981.

[10] W. Reisig: "*Petrinetze*", Springer, Berlin, 1982.

[11] G. Haus: "*Sistemi reali e rappresentazione formale: concetti primitivi per modelli*", International Culture Seminar on "*Ordine e disordine. Gerarchia oltre le due culture: nuovi modelli epistemologici*",

Università di Padova/Institut du Futur di Parigi/Stanford University, Padova (1986), pp. 302-307, DSE, Bologna, 1987.

[12] G. Haus, A. Rodriguez: "*Music Description and Processing by Petri Nets*", 1988 Advances on Petri Nets, LNCS, N. 340, pp.175-199, Springer, Berlin, 1989.

[13] V. E. Kotov: "*An Algebra for Parallelism based on Petri Nets*", MFCS 1978, Proc. 7th Symposium, Zakopane, Polonia, Springer, Berlin, 1978.

[14] R. Valk: "*Self-Modifying Nets, a Natural Extension of Petri Nets*", ICALP 1978, LNCS, N. 62, pp. 464-476, Springer, Berlin, 1978.

[15] E. Bianchi: "*Descrizione di partiture eseguibili su Macintosh mediante reti di Petri*", Computer Science Master Th., A.A. 85-86, Università degli Studi, Milano.

[16] A. Sametti: "*Un sistema per la sintesi di partuture musicali mediante esecuzione di reti di Petri*", Computer Science Master Th., A.A. 87-88, Università degli Studi, Milano.

[17] S. Scolaro: "*Modelli di partiture musicali mediante Reti di Petri: strumenti software e casi esemplari*", Computer Science Master Th., A.A. 89-90, Università degli Studi, Milano.

[18] G. Haus, A. Sametti: "*SCORESYNTH: a System for the Synthesis of Music Scores based on Petri Nets and a Music Algebra*", IEEE Computer, July 1991 (submitted).

[19] G. De Poli, G. Haus: "*Ingegneria del software ed informatica musicale*", Proc. A.I.C.A. Annual Conf., pp. 415-430, Università di Padova, 1982.

[20] S. W. Smoliar: "*A Parallel Processing Model of Musical Structures*", Ph. D. Thesis, Project MAC TR-74, M.I.T., Cambridge, Massachusetts, 1971.

[21] R. B. Dannenberg: "*A Functional Language for Real Time Control*", Communications ACM, Vol. 27, N. 8.

[22] J. A. Goguen: "*Complexity of Hierarchically Organized Systems and the Structure of Musical Experience*", UCLA Computer Science Dept. Quarterly, Vol. 3, N. 4, 1975.

[23] A. Bertoni, G. Haus, G. Mauri, M. Torelli: "*A Mathematical Model for Analyzing and Structuring Musical Texts*", Interface, Vol. 7, N. 1, pp. 31-44, Swets & Zeitlinger B.V., Amsterdam, 1978.