

INTERNATIONAL STANDARD

NORME INTERNATIONALE

GRAFCET specification language for sequential function charts

Langage de spécification GRAFCET pour diagrammes fonctionnels en séquence

IECNORM.COM : Click to view the full PDF of IEC 60848:2013



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2013 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,...).

It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available on-line and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Liens utiles:

Recherche de publications CEI - www.iec.ch/searchpub

La recherche avancée vous permet de trouver des publications CEI en utilisant différents critères (numéro de référence, texte, comité d'études,...).

Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

Just Published CEI - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications de la CEI. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (VEI) en ligne.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE

GRAFCET specification language for sequential function charts

Langage de spécification GRAFCET pour diagrammes fonctionnels en séquence

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XA

ICS 29.020

ISBN 978-2-83220-648-5

Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
1 Scope.....	8
2 Normative references.....	8
3 Terms and definitions	8
3.1 Terms in the GRAFCET	8
3.2 Terms, general purpose	10
4 General principles.....	10
4.1 Context.....	10
4.2 GRAFCET, a behaviour specification language	11
4.3 GRAFCET, short presentation.....	12
4.3.1 General	12
4.3.2 Structure.....	12
4.3.3 Elements for interpretation	12
4.4 Syntax rule	13
4.5 Evolution rules	14
4.5.1 General	14
4.5.2 Initial situation	14
4.5.3 Clearing of a transition	14
4.5.4 Evolution of active steps	14
4.5.5 Simultaneous evolutions	14
4.5.6 Simultaneous activation and deactivation of a step	14
4.6 Input events.....	14
4.6.1 General	14
4.6.2 Input events specification.....	15
4.7 Internal events.....	15
4.7.1 General	15
4.7.2 Internal events described by the step activation.....	15
4.7.3 Internal events described by the deactivation of a step	15
4.7.4 Internal events described by the clearing of a transition.....	15
4.8 Output modes	16
4.8.1 General	16
4.8.2 Continuous mode (assignation on state).....	16
4.8.3 Stored mode (allocation on event)	16
4.9 Application of the evolution rules.....	16
4.9.1 General	16
4.9.2 Non transient evolution	17
4.9.3 Transient evolution	17
4.9.4 Consequence of a transient evolution on the assignations	17
4.9.5 Consequence of a transient evolution on the allocations	18
4.10 Comparison between the two output modes	18
4.10.1 General	18
4.10.2 Determination of the value of the outputs	19
4.10.3 Analysis of the value of the outputs for a grafcet chart at a defined instant	19
4.10.4 Actions relative to transient evolution	19
4.10.5 Possible conflict on the value of the outputs	19

5	Graphical representation of the elements	19
6	Graphical representation of sequential structures	32
6.1	General	32
6.2	Basic structures	32
6.2.1	Sequence	32
6.2.2	Cycle of a single sequence.....	32
6.2.3	Selection of sequences	33
6.2.4	Step skip	33
6.2.5	Backward sequence skip.....	34
6.2.6	Activation of parallel sequences	34
6.2.7	Synchronization of sequences	34
6.2.8	Synchronization and activation of parallel sequences.....	35
6.3	Particular structures.....	36
6.3.1	Starting of a sequence by a source step	36
6.3.2	End of a sequence by a pit step	36
6.3.3	Starting of a sequence with a source transition.....	37
6.3.4	End of a sequence by a pit transition	38
7	Structuring.....	38
7.1	General	38
7.2	Partition of a grafcet chart.....	38
7.2.1	Connected grafcet chart.....	38
7.2.2	Partial grafcet	39
7.3	Structuring using the forcing of a partial grafcet chart.....	40
7.4	Structuring using the enclosure	41
7.5	Structuring using the macro-steps.....	43
Annex A (informative)	Example of the control of a press	45
Annex B (informative)	Example: Automatic weighing-mixing.....	46
Annex C (informative)	Relations between GRAFCET of IEC 60848 and the SFC of IEC 61131-3.....	52
Bibliography		54
Figure 1	– Graphical representation of the sequential part of a system.....	11
Figure 2	– Structure and interpretation elements used in a grafcet chart to describe the behaviour of a sequential part of the system defined by its input and output variables	13
Figure 3	– Example of grafcet with enclosures (including description)	43
Figure A.1	– Representation of the working press using a grafcet	45
Figure B.1	– Overview diagram of weighing-mixing system	46
Figure B.2	– Grafcet of a weighing-mixing involving only continuous actions	47
Figure B.3	– Grafcet of the weighing-mixing, involving continuous and stored actions	48
Figure B.4	– Grafcet of the weighing-mixing, divided into a global description using macro-steps and a description detailed by the macro-step expansions.....	49
Figure B.5	– Structuring with operating modes using forcing orders	50
Figure B.6	– Structuring with operating modes using enclosing step.....	51
Table 1	– Steps.....	20
Table 2	– Transitions.....	21
Table 3	– Directed links.....	22

Table 4 – Associated transition-conditions	23
Table 5 – Continuous actions	27
Table 6 – Stored actions	30
Table 7 – Comments associated with elements of a grafcet chart	31
Table 8 – Partial grafcet chart	39
Table 9 – Forcing of a partial grafcet chart	40
Table 10 – Enclosing steps.....	41
Table 11 – Macro-steps.....	44

IECNORM.COM : Click to view the full PDF of IEC 60848:2013

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**GRAFCET SPECIFICATION LANGUAGE
FOR SEQUENTIAL FUNCTION CHARTS****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 60848 has been prepared by the former subcommittee 3B: Documentation, of IEC technical committee 3: Information structures, documentation and graphical symbols.

This third edition cancels and replaces the second edition published in 2002 and constitutes a global technical revision with the extended definition of the concept of variables introducing: internal variable, input variable and output variable.

The text of this standard is based on the following documents:

FDIS	Report on voting
3/1135/FDIS	3/1138/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IECNORM.COM : Click to view the full PDF of IEC 60848:2013

INTRODUCTION

This International Standard is mainly aimed at people such as design engineers, maintenance engineers, etc., who need to specify the behaviour of a system, e.g. the control and command of an automation system, safety component, etc. This specification language should also serve as a communication means between designers and users of automated systems.

IECNORM.COM : Click to view the full PDF of IEC 60848:2013

GRAFCET SPECIFICATION LANGUAGE FOR SEQUENTIAL FUNCTION CHARTS

1 Scope

This International Standard defines the GRAFCET¹ specification language for the functional description of the behaviour of the sequential part of a control system.

This standard specifies the symbols and rules for the graphical representation of this language, as well as for its interpretation.

This standard has been prepared for automated production systems of industrial applications. However, no particular area of application is excluded.

Methods of development of a specification that makes use of GRAFCET are beyond the scope of this standard. One method is for example the "SFC language" specified in IEC 61131-3, which defines a set of programming languages for programmable controllers.

NOTE See Annex C for further information on the relations between IEC 60848 and implementation languages such as the SFC of IEC 61131-3.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

(void)

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

NOTE The definitions of the terms in 3.1 apply only in the context of the GRAFCET specification language.

3.1 Terms in the GRAFCET

3.1.1 action

GRAFCET language element associated with a step, indicating an activity to be performed on output or internal variables

3.1.2 directed link

GRAFCET language element indicating the evolution paths between steps by connecting steps to transitions and transitions to steps

3.1.3 grafcet chart

function chart using the GRAFCET specification language

¹ GRAFCET: GRAPhe Fonctionnel de Commande Etape Transition.

Note 1 to entry: The “grafcet chart” can, in short form, be called “grafcet”.

3.1.4

input event

event characterized by the change of at least one value of all input variables of the sequential part of the system

3.1.5

internal event

event characterized by an input event associated with the situation of the sequential part of the system

3.1.6

interpretation

part of the GRAFCET specification language enabling the linkage of:

- the input variables and the structure, by the means of the transition-condition; and
- the output variables and the structure, by the means of the actions

3.1.7

situation

state of the system described by the GRAFCET specification language and characterized by the active steps at a given instant

3.1.8

step

GRAFCET language element used for the definition of the state of the sequential part of the system

Note 1 to entry: A step can be active or inactive.

Note 2 to entry: The set of active steps represents the situation of the system.

3.1.9

transient evolution

evolution characterized by the clearing of several successive transitions on the occurrence of a single input event

3.1.10

transition

GRAFCET language element indicating a possible evolution of the activity between two or more steps

Note 1 to entry: The possible evolution is realised by clearing the transition.

3.1.11

transition-condition

GRAFCET language element associated with a transition indicating the result of a boolean expression

Note 1 to entry: The transition-condition can be either true or false.

3.1.12

variable

scalar quantity defined by its name and Boolean, numeric value

3.1.13

input variable

variable which may influence the behaviour described by the grafcet chart

EXAMPLE Boolean variable indicating the violation of a temperature limit.

Note 1 to entry: The variable may belong to the environment or to some other system component.

3.1.14

output variable

variable which may be influenced by the behaviour described by the grafcet chart

EXAMPLE Setpoint of a PID-controller.

Note 1 to entry: The variable may belong to the environment or to some other system component.

3.1.15

internal variable

variable used inside the grafcet chart and invisible for other system components and the environment

EXAMPLES Step variable X^* (symbol 2.1 of Table 1), step duration T^* (symbol 2.2 of Table 1), loop counter within a grafcet chart.

3.2 Terms, general purpose

3.2.1

chart

graph

graphical presentation describing the behaviour of a system, for example the relations between two or more variable quantities, operations or states

3.2.2

structure

part of the GRAFCET specification language enabling the description of the possible evolution between situations

3.2.3

system

set of interrelated elements considered in a defined context as a whole and separated from their environment

Note 1 to entry: Such elements may be material objects and concepts as well as their results (e.g. forms of organisation, mathematical methods, programming languages).

Note 2 to entry: The system is considered to be separated from the environment and from the other external systems by an imaginary surface, which cuts the links between them and the system.

Note 3 to entry: The language GRAFCET can be used to describe the logical behaviour of any kind of system.

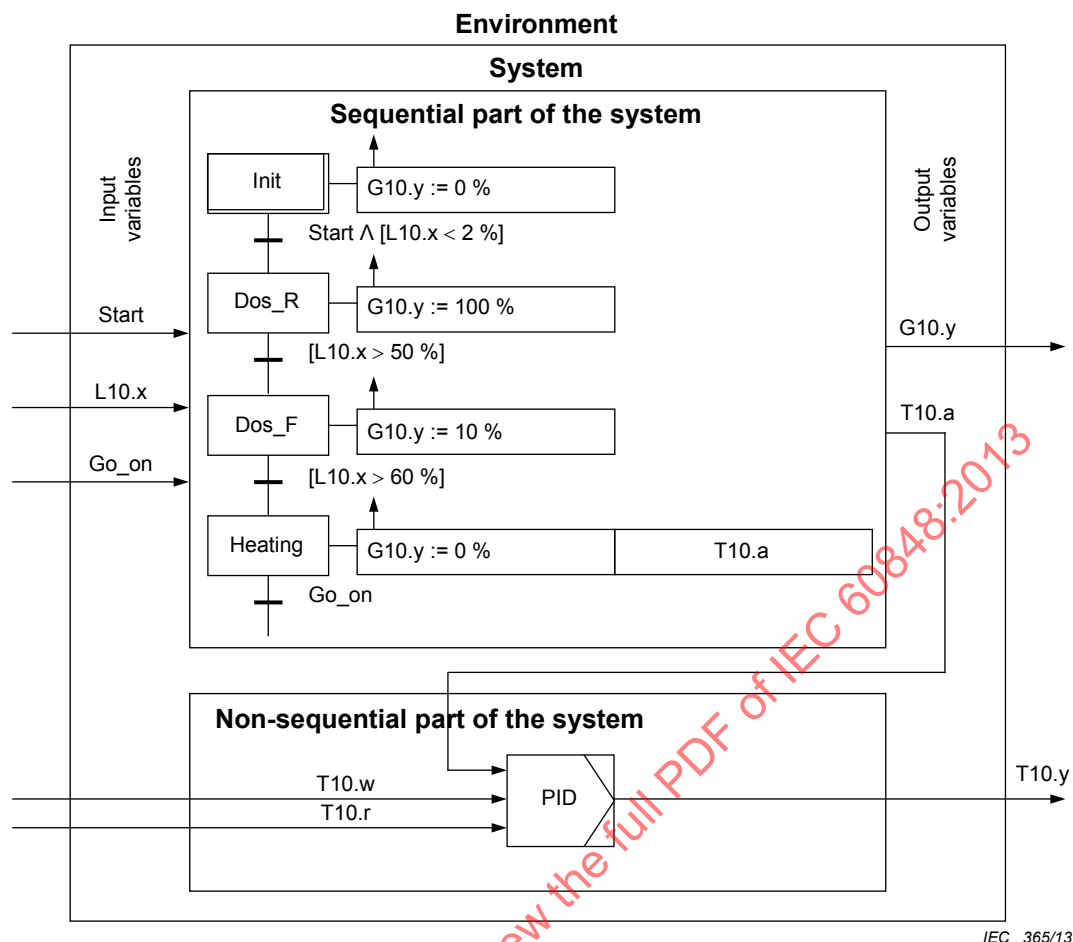
[SOURCE: IEC 60050-351:1998, 351-11-01]

4 General principles

4.1 Context

The implementation of an automated system requires, in particular, a description relating cause and effect. To do this, the logical aspect of the desired behaviour of the system will be described.

The sequential part of the system is the logical aspect of this physical system (see Figure 1). The behaviour indicates the way in which the output variables depend on the input variables. The object of the grafcet chart is to specify the behaviour of the sequential part of the systems.



L10.x tank level

T10.a temperature loop – automatic mode

T10.r temperature loop – measured value

G10.y dosing valve – position

T10.w temperature loop – setpoint

T10.y temperature loop – manipulated value

Figure 1 – Graphical representation of the sequential part of a system

4.2 GRAFCET, a behaviour specification language

The GRAFCET specification language enables a grafcet chart to be created showing the expected behaviour of a given sequential system. This language is characterized mainly by its graphic elements, which, associated with an alphanumerical expression of variables, provides a synthetic representation of the behaviour, based on an indirect description of the situation of the system.

The behaviour description on states is the following: the "monomarked" states correspond to the situations of the grafcet chart, which implies the uniqueness of the situation at a given instant. The states are connected to each other by means of an evolution condition, which allows the passage from one situation to another one to be described.

For reasons of convenience, the behaviour description based on states is better replaced by a description based on steps. In the grafcet chart several steps may be active simultaneously, the situation being then characterized by the set of active steps at the considered moment. The evolution of one set of steps to another is translated by one or several transitions, each characterized by:

- its preceding steps,
- its succeeding steps,

- its associated transition-condition.

NOTE These reasons lead to the syntax rule enforcing the alternation step-transition.

4.3 GRAFCET, short presentation

4.3.1 General

The GRAFCET specification language is used for the design of grafcet charts to provide a graphical and synthetic representation of the sequential systems behaviour. The representation (see Figure 2) distinguishes:

- the structure, which allows possible evolutions between the situations to be described, and
- the interpretation, which enables the relationship between input, output variables and the structure (evolution, assignation and allocation rules are necessary to achieve this interpretation).

Symbols related to GRAFCET elements representing steps in a process and links between the steps are presented and exemplified in Tables 1 to 4 in Clause 5.

4.3.2 Structure

The structure comprises the following basic items:

- Step (definition: 3.1.8, symbol 1). A step is either active or inactive, the set of the active steps of a grafcet chart at any given instant represents the situation of this grafcet chart at this instant.
- Transition (definition: 3.1.10, symbol 7). A transition indicates that an evolution of the activity between two or more steps may evolve. This evolution is realized by the clearing of the transition.
- Directed link (definition: 3.1.2, symbol 10). A directed link connects one or several steps to a transition, or a transition to one or several steps.

4.3.3 Elements for interpretation

The following elements are used for the interpretation:

- Transition-condition (definition: 3.1.11, symbol 13). Associated with each transition, the transition-condition is a logical expression which is true or false and which is composed of input variables and/or internal variables.
- Action (definition: 3.1.1). The action indicates, in a rectangle, what shall be done to the output or internal variable, either by assignation (continuous action, symbol 20), or allocation (stored action, symbol 26).

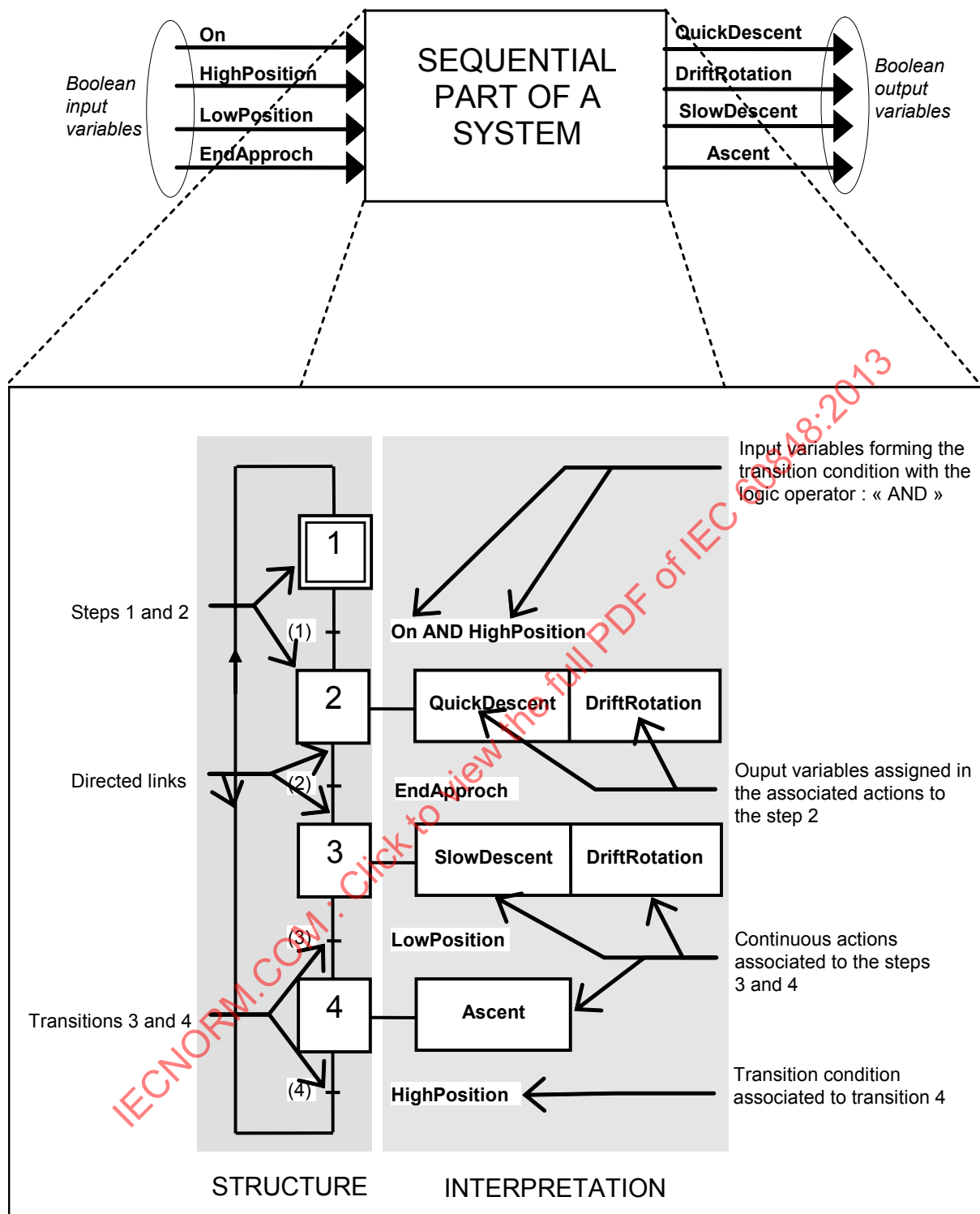


Figure 2 – Structure and interpretation elements used in a grafset chart to describe the behaviour of a sequential part of the system defined by its input and output variables

4.4 Syntax rule

Step transition and transition step alternation shall always be respected whatever the sequence.

Consequences:

- Two steps shall never be connected directly by a directed link;
- The directed link shall only connect a step to a transition or a transition to a step.

4.5 Evolution rules

4.5.1 General

As each situation is characterized by the set of active steps at a given instant, the grafcet evolution rules only affect the application, on the steps, of the evolution principle between the situations of the sequential part of the system.

4.5.2 Initial situation

The initial situation is the situation at the initial time. Therefore, it is described by the set of steps active at this time. The choice of the situation at the initial time depends on the methodology relating to the type of sequential part of the system considered.

Rule 1: The initial situation, chosen by the designer, is the situation at the initial time.

4.5.3 Clearing of a transition

Rule 2: A transition is said to be enabled when all immediately preceding steps linked to this transition are active. The clearing of a transition occurs:

- when the transition is enabled, and
- when its associated transition-condition is true.

4.5.4 Evolution of active steps

Rule 3: The clearing of a transition simultaneously provokes the activation of all the immediate succeeding steps and the deactivation of all the immediate preceding steps.

4.5.5 Simultaneous evolutions

The evolution between two active situations implies that no other intermediate situation is possible and the change from one representation of the situation by a set of steps to another representation is instantaneous.

Rule 4: Several transitions, which can be cleared simultaneously, are simultaneously cleared.

4.5.6 Simultaneous activation and deactivation of a step

If a step is included in the description of the preceding situation and in that of the following one, it can therefore only remain active.

Rule 5: If during the operation, an active step is simultaneously activated and deactivated, it remains active.

4.6 Input events

4.6.1 General

The evolution rules show that only a change in the values of the input variables may cause the evolution of the grafcet chart. This change called "input event" shall be defined by the preceding value and the succeeding value of all the input variables to characterise this single event. In practice, a set of input events is specified only by the characterised state change (rising edge or falling edge) of one or several Boolean input variables.

NOTE The rising edge of a logical variable, indicated by the sign "↑" in front of a Boolean variable, indicates that this rising edge is only true for the change from value 0 to value 1 of the variable concerned. The falling edge of a

logical variable noted by the sign " $\downarrow$ " in front of a Boolean variable, indicates that this falling edge is only true for the change from value 1 to value 0 of the variable concerned.

It is said that "the event occurs" at the date of the change of state of the input variables which characterize it.

4.6.2 Input events specification

The input events specification is implemented by a logical expression of one or several characteristic variables, often in a transition-condition. More rarely, it may also directly specify an internal event (see 4.7).

$\uparrow a$ EXAMPLE 1:

The expression " $\uparrow a$ " describes the set of all input events for which the preceding value of the input variable "a" is 0 and its succeeding value is 1, regardless of the value of the other input variables of the system.

$a \cdot \uparrow b$ EXAMPLE 2:

The expression " $a \cdot \uparrow b$ " describes the set of all input events for which the succeeding value of the input variable "a" is 1, and the preceding value of the input variable "b" is 0 and its succeeding value is 1, regardless of the value of the other input variables of the system.

a EXAMPLE 3:

The expression " a " describes the sets of all input events for which the succeeding value of the input variable "a" is 1, regardless of the value of the other input variables of the system.

NOTE Used in a transition-condition, this expression could lead to a transient evolution (see 3.12).

4.7 Internal events

4.7.1 General

Only certain input events could occur from a given situation. The connection between a situation and input event, which may occur from this situation, is called internal event (see 3.6). This notion is mainly used by the designer to condition an output allocation to a set of internal events (see 4.8.3). The description of a set of internal events is realized by one of the following ways.

4.7.2 Internal events described by the step activation

The step activation, noted graphically (symbol 27), describes the set of internal events each of which has this step activation as a consequence.

4.7.3 Internal events described by the deactivation of a step

The graphically noted deactivation of a step (symbol 28) describes the set of the internal events each of which have this step deactivation as consequence.

4.7.4 Internal events described by the clearing of a transition

The graphically noted clearing of a transition (symbol 29) describes the set of internal events each of which have the clearing of this transition as consequence.

4.8 Output modes

4.8.1 General

The actions enable links to establish the connection between the evolution of the grafcet chart and the outputs. Two output modes, continuous mode or stored mode, describe how the outputs depend on the evolution and on the system inputs.

4.8.2 Continuous mode (assignment on state)

In the continuous mode, the association of an action with a step indicates that an output variable has a true value if the step is active and if the assignment condition is verified. The assignment condition is a logical expression of the input variables and/or the internal ones (see symbol 22). If one of the conditions is not met and provided that no other action relating to the same output meets the conditions, the output variable concerned takes the false value.

Assignment refers to imposing the values of the output variables (true or false).

The set of the local assignment (relating to the active steps at a given instant) defines the assignment of all the output variables for this situation.

Assignment rule: for a given situation, the value of the outputs relating to the continuous actions is assigned:

- to the true value, for each output relating to the actions associated with active steps and for which the assignment conditions are verified,
- to the false value, for the other outputs (which are not assigned to the true value).

4.8.3 Stored mode (allocation on event)

In the stored mode, the association of an action to internal events is used to indicate that an output or internal variable takes and maintains the enforced value if one of these events occurs.

Explicit representations are necessary to describe the association of the actions with the events (activation step, deactivation step, clearing of a transition, etc.).

The value of an output or internal variable relating to a stored action remains unchanged until a new specified event modifies its value.

Allocation refers to storing, at a considered moment, a determined value affected to an output or internal variable.

Allocation rule: the value of an output or internal variable, relating to a stored action and associated to an event, is allocated to the indicated variable, if the specified internal event occurs; the value of this variable is false (Boolean variable) or null (numeric variable) at the initialisation.

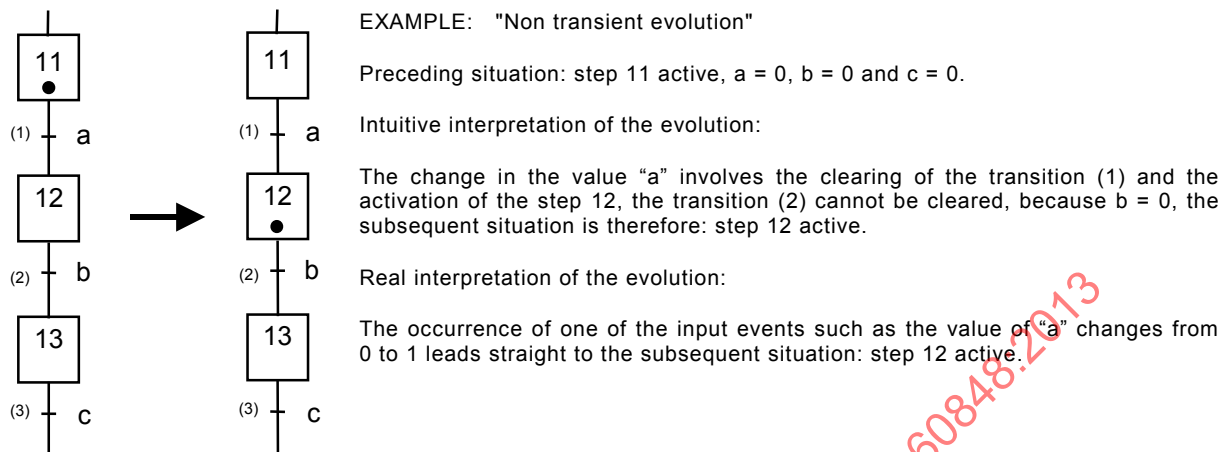
4.9 Application of the evolution rules

4.9.1 General

Intuitive interpretation of the evolution, called “step by step”, designates the progressive way which allows, on the occurrence of an input event and from the preceding situation, to determine the succeeding situation of this event, by the successive application of the evolution rules on each transition. The interpretation facility is a device to enable an indirect specification of the evolution, but the designer shall take care that the clearing of the transitions on this path does not involve the effective activation of the intermediate situations.

4.9.2 Non transient evolution

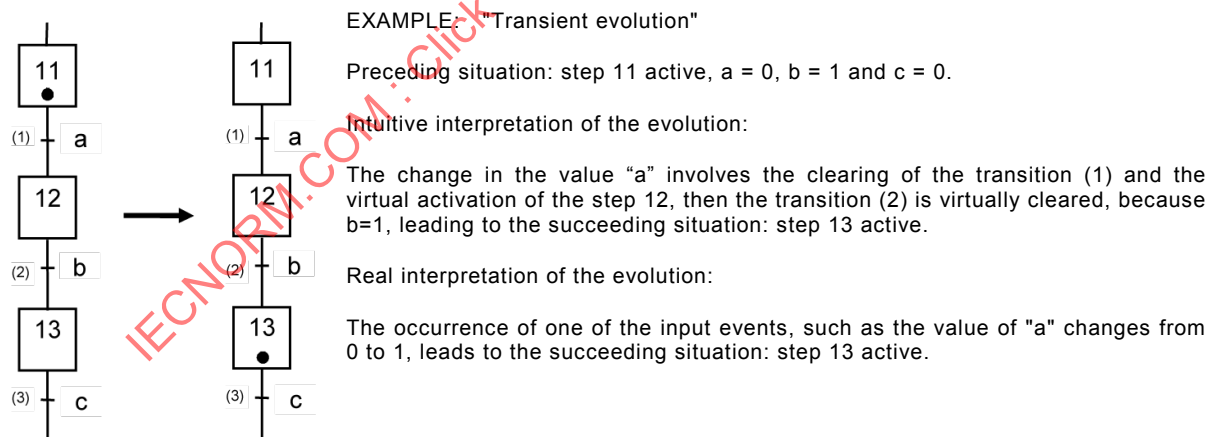
In general, the evolution is non-transient, which means that the input event only leads to one evolution stage (the simultaneous clearing of one or more transitions).



4.9.3 Transient evolution

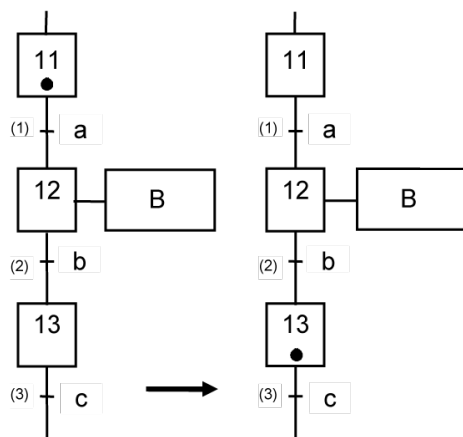
In some cases, the application of the evolution rules can lead to successively clearing some transitions (in several evolution stages) if the transition-conditions associated with the subsequent transitions are already true, when the first transitions considered are cleared. The corresponding description, referred to as transient, uses the path taken to indicate how to move from a preceding situation to a succeeding situation (see 3.9).

The corresponding intermediate steps, referred to as unstable are not activated, but we consider that they have been "virtually" activated and deactivated along the intuitive evolution path, as well as for the corresponding transitions which have been "virtually" cleared.



4.9.4 Consequence of a transient evolution on the assignments

The assignation of an output value by a continuous action associated with a step, which is an unstable step in the case of a transient evolution, is not effective, since the step is not really activated (see 4.8.2).



EXAMPLE: "Continuous action associated with an unstable step"

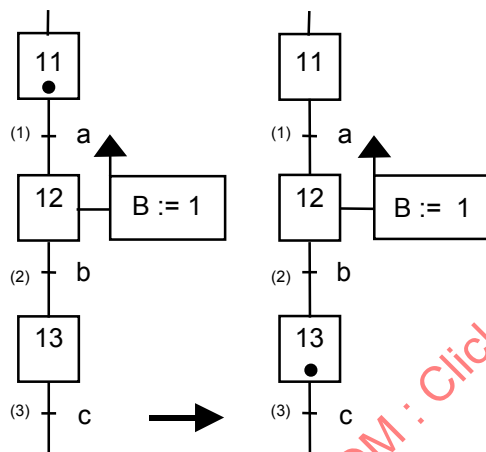
Preceding situation: step 11 active, $a = 0$, $b = 1$ and $c = 0$.

The occurrence of one of the input events such as the value of "a" changes from 0 to 1, leads straight to the subsequent situation: step 13 active.

The preceding situation (step 11 active) and the succeeding situation (step 13 active) assign the value 0 to the output variable B. The unstable step 12 being not really activated, the assignation of B to the value 1 is not effective on the transient evolution.

4.9.5 Consequence of a transient evolution on the allocations

The allocation to a determinate value of an output by a stored action (symbol 26) associated to a step, which is an unstable step in the case of a transient evolution, is effective since this allocation is associated with the events releasing this evolution (see 4.8.3).

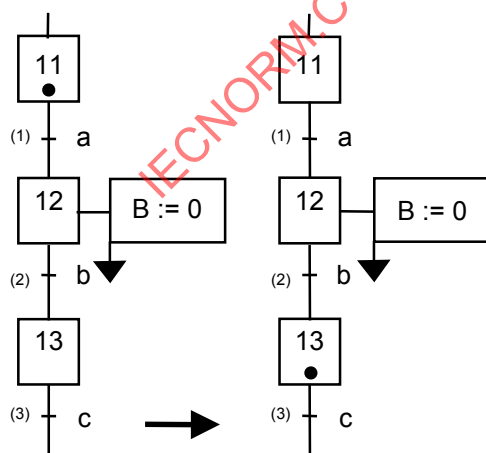


EXAMPLE 1: "Stored action associated with the activation of an unstable step"

Preceding situation: step 11 active, $a = 0$, $b = 1$ and $c = 0$.

The occurrence of one of the input events such as the value of "a" changes from 0 to 1, leads straight to the subsequent situation: step 13 active.

The allocation of the value 1 to the output variable B is realized on the occurrence of one of the input events having the real or the virtual activation of the step 12 as consequence.



EXAMPLE 2: "Stored action associated with the deactivation of an unstable step"

Preceding situation: step 11 active, $a = 0$, $b = 1$ and $c = 0$.

The occurrence of one of the input events such as the value of "a" changes from 0 to 1, leads straight to the subsequent situation: step 13 active.

The allocation of the value 0 to the output variable B is realized on the occurrence of one of the input events having the real or the virtual deactivation of the step 12 as consequence.

4.10 Comparison between the two output modes

4.10.1 General

The choice of the output mode depends on the practice and methodology used. However, the designers' attention is drawn to the important differences between the two modes.

4.10.2 Determination of the value of the outputs

Depending on the chosen mode the determination of the value of the outputs are described as:

- in continuous mode, all the outputs are assigned according to the situation, to the true value for the outputs explicitly indicated in the actions associated to the active steps, and to the false value for the other ones which are implicitly set by omission (see assignation rule, 4.8.2);
- in the stored mode, only the considered outputs are modified according to the indicated value, the other stored values of the outputs remain unchanged (see allocation rule, 4.8.3).

4.10.3 Analysis of the value of the outputs for a grafcet chart at a defined instant

Depending on the chosen mode the analysis of the value of the outputs are described as:

- in the continuous mode, the knowledge of the situation and the value of the inputs is sufficient to determine the value of the outputs (see 4.8.2);
- in the stored mode, the knowledge of the situation and the value of the inputs is not sufficient, the preceding evolutions shall also be known to determine the value of the outputs (see 4.8.3).

4.10.4 Actions relative to transient evolution

Depending on the chosen mode the actions relative to transient evaluation are described as:

- in the continuous mode, the actions associated with an unstable step are not taken into consideration because this step is not activated (see 4.9.2);
- in the stored mode, the actions associated with events and in relation with a transient evolution are taken into consideration because the triggered events releasing this evolution occur (see 4.9.3).

4.10.5 Possible conflict on the value of the outputs

Depending on the chosen mode the possible conflict on the value of the outputs are managed as:

- in the continuous mode, the assignation principles ensure every assignation conflict on the particular output to be avoided;
- in the stored mode, the allocation rules do not allow the possible assignation conflicts on a same output to be avoided. The designer shall ensure that two contradictory allocations cannot occur simultaneously.

NOTE 1 Both output modes can be used in one specification in GRAFCET, but the value of an output variable is determined either by assignation or by allocation. The specification of an allocation to an output variable (stored mode), excludes this output variable of any assignation (continuous mode).

NOTE 2 Clause 5 gives the graphic symbols which enable the stored actions (indicated by explicit representation according to the set of specified events) to be distinguished from the continuous ones (indicated by absence of any representation).

NOTE 3 In the frequent case of the specification of control system behaviour, the current industrial practice forces the employment of the continuous mode for all the Boolean outputs to the actuators, and the stored mode for describing internal control tasks. These tasks, such as the incrementation of a counter, or the modification of the value for a numerical register, refer to internal variables, which are not necessarily Boolean ones. The internal tasks associated with the stored actions, as well as the calculation of expressions associated with transition-conditions, are not described in the present standard, but are associated by the use of the logical description of the grafcet evolutions.

5 Graphical representation of the elements

The elements of GRAFCET have their own symbolic representation which, when correctly associated, enable clear and synthetic function-charts to be implemented.

NOTE 1 Only the global representation of the symbols is imposed; dimensions and details (thickness of lines, font of characters, etc.) are left up to the users.

NOTE 2 The stippled representation indicates the context of the symbol.

Table 1 – Steps

No.	Symbol	Description
[1]		Step: At a given moment, a step is either active or inactive. The set of active steps defines the situation of the given system at the considered instant. The height-width ratio of the rectangle is arbitrary, although a square is recommended. For the purposes of identification, the steps shall have a label, for example, alphanumerical. The label assigned to the step shall replace the asterisk at the upper half of the general symbol. EXAMPLE 1: "Step 2"  EXAMPLE 2: "Step 3 represented in its active state"  NOTE It could be useful to indicate which steps are active at a given instant by marking these steps with a dot. This dot is not part of the step symbol and is only used for explanatory purposes.
[2.1]	X*	Step variable: The active or inactive state of the step may be represented by the logical values "1" or "0" respectively of a Boolean variable X*, in which the asterisk * shall be replaced by the label of the relevant step. EXAMPLE: "Step variable of the step 8" X8
[2.2]	T*	Step duration: The duration of an active step may be represented by the value of a time variable T*, in which the asterisk * shall be replaced by the label of the relevant step. EXAMPLE: "Step duration of step fill" TFill.
[3]		Initial step: This symbol means that this step participates in the initial situation. NOTE 1 The rules of symbol 1 apply. NOTE 2 An initial step could be "unstable", see 4.9.3. EXAMPLE: "Initial step 12" 
[4]		Enclosing step: This symbol indicates that this step contains other steps referred to as enclosed steps. NOTE 1 The rules of symbol 1 apply. NOTE 2 The properties and the examples of the use of the enclosing step are given in 7.4.
[5]		Initial enclosing step: This symbol means that this enclosing step participates in the initial situation. NOTE An initial enclosing step contains at least one enclosed initial step.
[6]		Macro-step: Unique representation of a detailed part of the function-chart referred to as the expansion of the macro-step. NOTE The properties and the examples of the use of the macro-step are given in 7.5.

Table 2 – Transitions

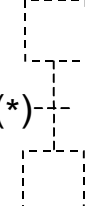
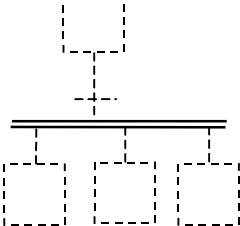
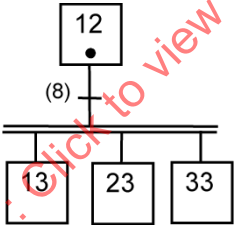
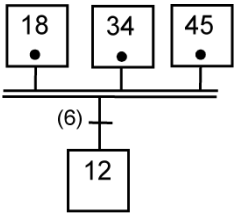
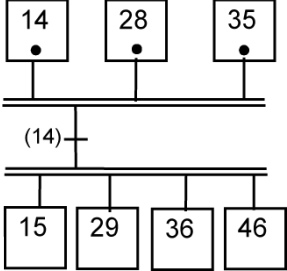
No.	Symbol	Description
[7]		Transition from one step to another: A transition is represented by a line perpendicular to the link joining two steps. NOTE 1 The transition is enabled when the immediate preceding step is active (see the evolution rule No. 2, 4.5.3). NOTE 2 <i>Only one transition</i> is ever possible between two steps (see 4.4). NOTE 3 It is possible, for graphical representation reasons, to place transitions on horizontal directed links (see Figure B5, partial grafcet G1). NOTE 4 The symbolism of transitions is not subject of this standard. Transitions can be described by plain text, Boolean expressions, logic charts, etc.
[8]		Transition designation: The transition may have a designation, generally placed to the left, which should not be mistaken for the associated transition-condition. An alphanumerical designation for the transition shall replace the asterisk.
[9]		Synchronization preceding and/or succeeding a transition: When several steps are connected to the same transition, the directed links from and/or to these steps are grouped, to succeed or precede the synchronization symbol represented by two parallel horizontal lines. NOTE The reference for the synchronization symbol is 9.2.2.5 of ISO 5807:1985.  Diagram for Example 1: A box labeled 12 with a dot inside is connected by a vertical line to a transition symbol (8) represented by a vertical line with a crossbar. From the transition, three vertical lines branch out to boxes labeled 13, 23, and 33, each with a dot inside. EXAMPLE 1: Transition from one step (12) to several (13, 23, 33). The transition (8) is enabled when the step 12 is active.  Diagram for Example 2: Three boxes labeled 18, 34, and 45, each with a dot inside, are connected by vertical lines to a synchronization symbol (6) represented by two parallel horizontal lines. From the synchronization symbol, a vertical line leads down to a box labeled 12 with a dot inside. EXAMPLE 2: Transition from several steps (18, 34, 45) to one step (12). The transition (6) is only enabled when all preceding steps are active.  Diagram for Example 3: Three boxes labeled 14, 28, and 35, each with a dot inside, are connected by vertical lines to a transition symbol (14) represented by a vertical line with a crossbar. From the transition, four vertical lines branch out to boxes labeled 15, 29, 36, and 46, each with a dot inside. EXAMPLE 3: Transition from several steps (14, 28, 35) to several steps (15, 29, 36, 46). The transition (14) is only enabled when all preceding steps are active.

Table 3 – Directed links

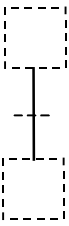
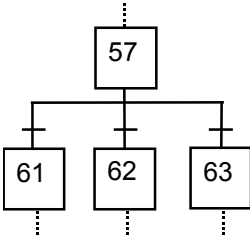
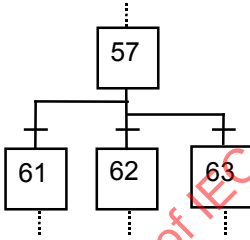
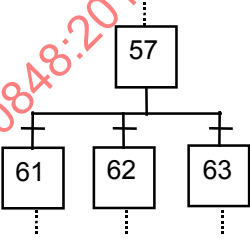
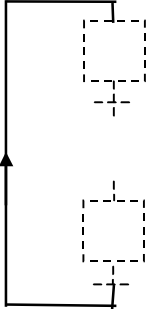
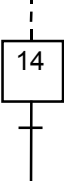
No.	Symbol	Description
[10]		Directed link from top to bottom: The evolution paths between the steps are indicated by directed links connecting steps to transitions and transitions to steps. Directed links are horizontal or vertical. Diagonal links are only permitted in those rare cases where they improve the clarity of the chart. Crossovers of vertical and horizontal links are permitted if no relationship exists between those links. Accordingly such crossovers shall be avoided when the links are related to the same evolution. EXAMPLE: The three representations are permitted but the representations 2 and 3 are recommended to avoid misunderstanding between links with and without relationship.  (1)  (2)  (3)
[11]		Directed link from bottom to top: By convention, the direction of the evolution is always from top to bottom. Arrows shall be used if this convention is not respected or if their presence enables a clearer understanding.
[12]		Linked label: If a directed link has to be broken (for example in complex charts or when one chart covers several pages) the number of the destination steps and the number of the page on which it appears, shall be indicated. The asterisk shall be replaced by the linked label. EXAMPLE: Evolution to step 83 of page 13.  Step 83 Page 13

Table 4 – Associated transition-conditions

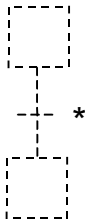
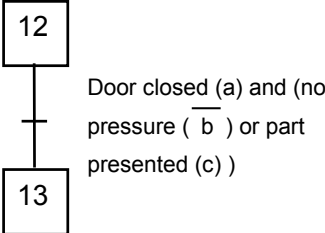
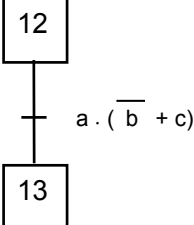
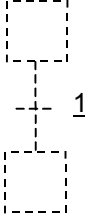
No.	Symbol	Description
[13]		Transition-condition: A logical proposition, called a transition-condition, which can be either true or false, is associated with each transition. If a corresponding logical variable exists, it is equal to 1 when the transition-condition is true or equal to 0 when the transition-condition is false. The logical proposition forming the transition-condition comprises one or several Boolean variables, (input variable, step variable, predicate value, etc.). The asterisk shall be replaced by the description of the transition-condition in the form of text, of a Boolean expression, or by using graphical symbols.   EXAMPLE 1: Transition-condition described by a text. EXAMPLE 2: Transition-condition described by a Boolean expression.
[14]		Transition-condition always true: The notation "1" means that the transition-condition is always true. NOTE In this case, the evolution is to be transient (see 4.9.3), the clearing of the transition is only conditioned by the activity of the preceding step.

Table 4 (continued)

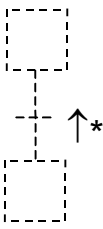
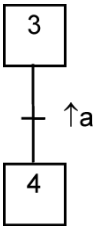
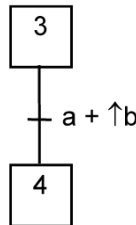
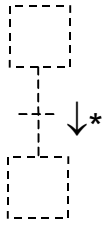
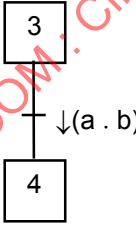
No.	Symbol	Description
[15]		Rising edge of a logical variable: The notation "$\uparrow$" means that the transition-condition is only true at the change of the state of the variable $*$ (rising edge: changing from value 0 to value 1, see the note in 4.6). This symbol is general and applies to all logical propositions, either for an elementary variable or for a set of several Boolean variables.  EXAMPLE 1: The associated transition-condition is only true when a changes from state 0 to state 1. NOTE By applying the evolution rule No. 2, the transition is only cleared on a rising edge of a after the transition has been enabled by the activity of step 3.  EXAMPLE 2: The associated transition-condition is true only when a is true or when b changes from state 0 to state 1.
[16]		Falling edge of a logical variable: The notation "$\downarrow$" means that the transition-condition is only true on the change of the state of the variable $*$ (falling edge: changing from value 1 to value 0, see the note in 4.6). This symbol is general and applies to all logical propositions, either for an elementary variable or for a set of several Boolean variables.  EXAMPLE: The associated transition-condition is true only when the logical product "$a \cdot b$" changes from state 1 to state 0.

Table 4 (continued)

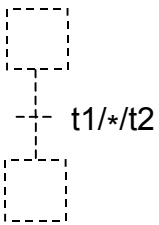
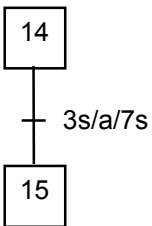
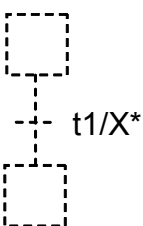
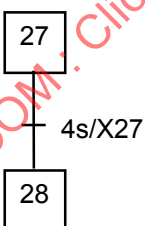
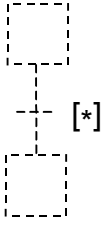
No.	Symbol	Description
[17]		Time dependent transition-condition: The notation " t1/*/t2 " indicates that the transition-condition is true only after a time t1 from the occurrence of the rising edge (↑*) of the time limited variable and becomes false again after a time t2 from the occurrence of the falling edge (↓*). The asterisk shall be replaced by the time-delayed variable, for example a step variable or an input variable. t1 and t2 shall be replaced by their real value expressed in the selected time unit. The time-delayed variable shall remain true for a period equal to or greater than t1 for the transition-condition be true. NOTE This notation is that of the delay element defined by the standard IEC 60617-S01655 (2004-09).  EXAMPLE: The transition-condition is true 3 s after the change of "a" from state 0 to the state 1, it becomes false 7 s after the change of "a" from state 1 to the state 0.
[18]		Usual simplification of symbol 17: Current use is to delay the step variable by a time t2 equal to zero, then, the transition-condition becomes false on deactivation of the step * that activated the delay. The asterisk shall be replaced by the label of the step which is required to be delayed. The time delayed step shall remain active during a time equal to or greater than t1 for the transition-condition to be true. This notation can be used when the time-delayed step is not the preceding step of the transition.  EXAMPLE: The transition-condition will be true during 4 s after the activation of step 27, and will be false with the clearing of the transition which deactivates the preceding step. In this case, the duration of the activity of the step 27 is 4 s.

Table 4 (continued)

No.	Symbol	Description
[19]		Boolean value of a predicate: "[*]" indicates that the Boolean value of the predicate constitutes the transition-condition variable. Therefore, when the assertion * is verified, the predicate has value of 1, otherwise the predicate has a value of 0. The asterisk shall be replaced by the assertion, which shall be tested. The Boolean variable of the predicate can be associated with other logical variables to constitute a logical proposition of transition-condition. EXAMPLE 1: The transition-condition is true when the assertion "C1=3" is verified. EXAMPLE 1a: The transition-condition is true when the current value of the counter C1 is equal to 3. NOTE The form of the assertion is not imposed; for example a literal language can be used. EXAMPLE 2: The transition-condition is true when the assertion "$t > 8\text{ °C}$" is verified and when the Boolean variable k has a value of "1", that means, when the temperature t is higher than the value 8 °C and when the high level k is reached. EXAMPLE 3: The transition-condition is true when the Boolean variable "b" has a value of 1 or when the assertion "$R1 \neq 24$" is verified, that means when the part is at the place b, or when the register R1 has not yet reached the value of 24.

Symbols representing action are presented and exemplified in Table 5 and Table 6 below. Actions can be of type continuous actions (see Table 5) or stored actions (see Table 6).

A stored action has a label (symbol 26) situated in the rectangle which describes how the output variable is allocated to a determinate value according to the allocation rule (see 4.8.3).

The event specification associated with the stored action is necessary to indicate when the corresponding output allocations occur (see allocation rule 4.8.3). Four means of description (symbols 27 to 29) allow the easy specification of different sets of internal events associated with the stored actions.

Table 5 – Continuous actions

No.	Symbol	Description
[20]		Continuous action: A continuous action is necessarily associated with a step. Several actions can be associated with one step. The height-width ratio is arbitrary although a rectangle of the same height as the step is recommended. In the absence of an explicit symbolisation of a stored action (symbols 27 to 29), the general rectangular symbol associated with a step always designates a continuous action.
[21]		Assignment label of an output: Each action shall have a label inside the rectangle, which refers to this action. The label of a continuous action is the designation of the output variable assigned to the true value according to the assignment rule (see 4.8.2). The asterisk shall be replaced by the wording of the output variable. The textual expression of the label can take an imperative or indicated form, the only important point is the reference to the output. The order in which the actions are represented does not imply any sequence between the actions. EXAMPLE 1: Different forms, literal or symbolic, of an action label which refers to the output when the value is true, will provoke valve 2 to open. 4 To open valve 2 4 valve 2 4 YV2 EXAMPLE 2: Different representations (1, 2, 3, 4) of the association of several actions at one step. (1) 6 A B C (2) 6 A B C (3) 6 A B C (4) 6 A B C NOTE The four representations are strictly equivalent. Representation (2) and (4) can be considered respectively as simplifications of the representation (1) and (3).

Table 5 (continued)

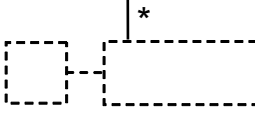
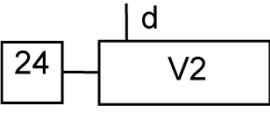
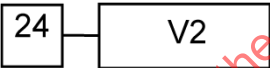
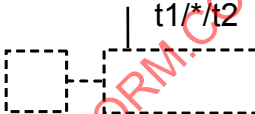
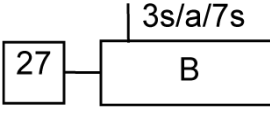
No.	Symbol	Description
[22]		Assignment condition: A logical proposition, called an assignment condition, which can be true or false, influences any continuous action. The absence of notation indicates that the condition is always true. The assignment condition description in text format or a Boolean expression between the input variables and/or the internal variables shall replace the asterisk. This assignment condition shall never include an edge of variable (see symbols 15 and 16), because the continuous action is of course not memorised, an assignment on event having no meaning (see 4.8.3). EXAMPLE 1: Output V2 is assigned to the true value when step 24 is active and when the assignment condition d is true. In the opposite case, output V2 is assigned to the false value. In other words (as a Boolean equation): $V2 = X24 \cdot d$  NOTE X24 is the step variable which reflects the activity of step 24. EXAMPLE 2: Output V2 is assigned to the true value when step 24 is active (the assignment condition is always true). In the opposite case, output V2 is assigned to the false value. In other words (as a Boolean equation): $V2 = X24$ 
[23]		Time dependent assignment condition: The notation "t1/*t2" indicates that the assignment condition is true only after a time t1 from the occurrence of the rising edge (↑*, see symbol 15) of the timed variable * and becomes false again after a time t2 from the occurrence of the falling edge (↓*, see symbol 16). The asterisk shall be replaced by the timed variable, for instance a step variable or an input variable. t1* and t2 shall be replaced by their real value expressed in the selected time unit. The limited variable shall remain true for a time equal to or greater than t1 for the assignment condition to be true. NOTE This notation is that of the delay element defined by IEC 60617-S01655 (2004-09). EXAMPLE: The assignment condition is true only 3 s after "a" changes from state "0" to state "1", and false 7 s after "a" changes from state "1" to state "0".  The value of output B depends on the activity of step 27 and on the value of the assignment condition (see assignment rules 4.8.2).

Table 5 (continued)

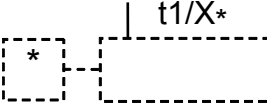
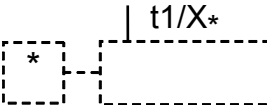
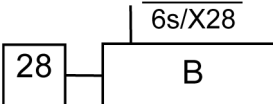
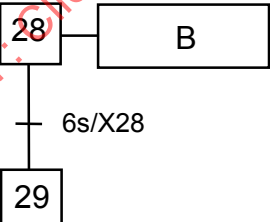
No.	Symbol	Description
[24]		Delayed action: The delayed action is a continuous action in which the assignation condition is true only after a time t_1 specified from the activation of the associated step *, with the objective of delaying the assignation to the true value of the corresponding output. EXAMPLE: Output B is assigned to the true value when 3 s have elapsed since the activation of step 27.  NOTE If the step 27 activity time is less than 3 s, then the output B variable is not assigned to the true value.
[25]		Time limited action: The time limited action is a continuous action in which the assignation condition is true for a period of time t_1 specified from the activation of the associated step *, for limiting the duration of the assignation to the true value of the corresponding output. EXAMPLE 1: Output B is only assigned to the true value for 6 s from the activation of step 28.  NOTE If the step 28 activity time is less than 6 s, the output B variable is assigned to the true value only during the step 28 activity time. Equivalent representation: The simplified delay operator can be used in the associated transition-condition for the succeeding step to limit the allocation time of the true value to the corresponding output (see symbol 18). EXAMPLE 2: Equivalent representation of the example 1 with the symbol 18. Output B is only assigned to the true value for 6 s from the activation of step 28. 

Table 6 – Stored actions

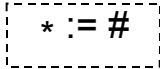
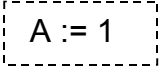
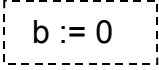
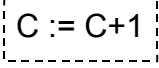
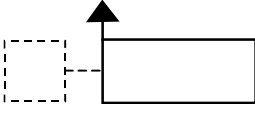
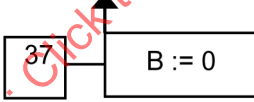
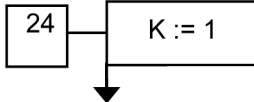
No.	Symbol	Description
[26]		Allocation of the value # to a variable *: The wording indicates, for a stored action, the setting to the value # of a variable * when one of the events associated with the action occurs (see allocation rule 4.8.3). The stored action supporting this allocation shall be associated with the internal events specification (symbols 27 to 29). The allocation can be described textually within the action rectangle.    EXAMPLE 1: Set the value of a Boolean variable A to true. The wording "A:= 1 " describes the allocation of the value 1 to a Boolean variable A when one of the events associated with the action occurs. EXAMPLE 2: Set the value of a Boolean variable b to false. The wording " b:= 0 " describes the allocation of the value 0 to a Boolean variable b when one of the events associated with the action occurs. EXAMPLE 3: Incrementation of a counter The wording "C:= C+1" describes the allocation of the value C+1 to a numeric variable C when one of the events associated with the action occurs.
[27]		Action on activation: An action on activation is a stored action associated with the set of the internal events, which have, for each one, the linked step activation as consequence. The traditional representation of the action by a rectangle is completed, on the left side, by an arrow symbolising the activation of the step.  EXAMPLE: The Boolean variable B is allocated to the value 0 when one of the events, leading to the activation of step 37, occurs.
[28]		Action on deactivation: An action on deactivation is a stored action associated with the set of the internal events, which have, for each one, the linked step deactivation as consequence. The traditional representation of the action by a rectangle is completed, on the left side, by an arrow symbolizing the deactivation of the step.  EXAMPLE: The Boolean variable K is allocated to the value 1 when one of the events, represented by the deactivation of step 24, occurs.

Table 6 (continued)

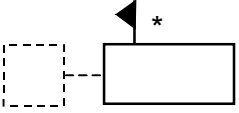
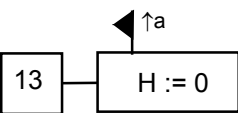
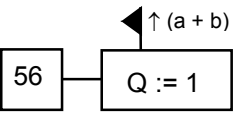
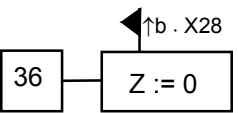
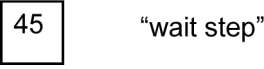
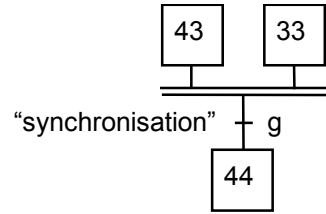
No.	Symbol	Description
[29]		Action on event: An action on event is a stored action associated with each of the internal events described by the expression * on condition that the step, with which the action is connected, is active. The traditional representation of the action by a rectangle is completed, on top, by a symbol indicating that the action is conditioned by the occurrence of one of the internal events specified by the expression *. It is recommended that the logical expression *, which shall describe a set of internal events, is made up of one or more input variable edges.  13 — H := 0 EXAMPLE 1: The Boolean variable H is allocated to the value 0 when one of the events, represented by "↑a", occurs and simultaneously, step 13 is active.  56 — Q := 1 NOTE The combination between the set of the input events, represented by the expression "↑a", and the step 13 activity represents in fact a set of internal events (see definition 3.6). EXAMPLE 2: The Boolean variable Q is allocated to the value 1 when one of the events, represented by "↑(a + b)", occurs and simultaneously, step 56 is active.  36 — Z := 0 EXAMPLE 3: The Boolean variable Z is allocated to the value 0 when one of the events, represented by "↑b", occurs and simultaneously, steps 36 and 28 are active.

Table 7 contains comments associated with GRAFCET elements.

Table 7 – Comments associated with elements of a grafcet chart

No.	Symbol	Description
[30]	"*"	Comment: A comment concerning the graphic elements of a function-chart shall be placed between inverted commas (quotation mark). The asterisk shall be replaced by the comment.  45 "wait step" EXAMPLE 1: Comment "wait step" referring to step 45.  28 — P.P. "punch part" EXAMPLE 2: Comment "punch part" referring to the action associated with step 28.  "synchronisation" — g EXAMPLE 3: Comment "synchronisation" referring to a transition.

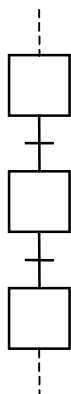
6 Graphical representation of sequential structures

6.1 General

The designer can construct grafcet charts using different distinctive structures, subject to strict application of the syntax rule concerning step/transition alternation.

6.2 Basic structures

6.2.1 Sequence



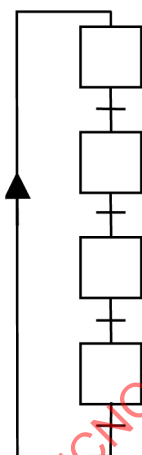
A sequence is a succession of steps such that:

- each step, except the last one, has only one succeeding transition,
- each step, except the first one, has only one preceding transition enabled by a single step of the sequence.

NOTE 1 The sequence is said to be "active" if at least one of its steps is active. The sequence is said to be "inactive" when none of its steps is active.

NOTE 2 A sequence may include any number of steps.

6.2.2 Cycle of a single sequence



The case of a looped sequence such that:

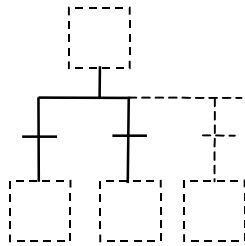
- each step has only one succeeding transition,
- each step has only one preceding transition enabled by a single step of the sequence.

NOTE 1 A cycle of a single sequence may constitute a partial grafcet (see 7.2.2).

NOTE 2 A cycle of a single sequence shall satisfy at least one of the following conditions to allow the activation of its steps:

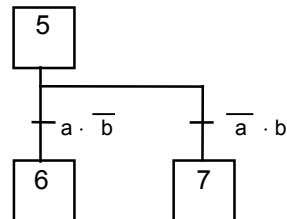
- to have at least one initial step,
- to be submitted by a forcing order from a partial grafcet at a higher level (see 7.3),
- to belong to one of the encapsulations of an enclosing step (see 7.4).

6.2.3 Selection of sequences

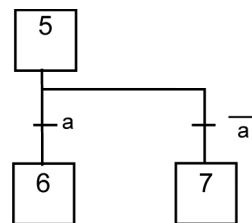


The selection of sequences shows a choice of evolution between several sequences starting from one or several steps. This structure is represented by as many simultaneously enabled transitions as possible evolutions.

NOTE Exclusive activation of a selected sequence is not guaranteed from the structure. The designer should ensure that the timing, logical or mechanical aspects of the transition-conditions are mutually exclusive.

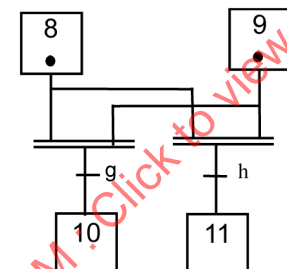


EXAMPLE 1: The exclusion between the sequences is achieved by the logical exclusion of the two receptivities. If "a" and "b" are simultaneously true when step 5 is active, no transition may be cleared.



EXAMPLE 2: Priority sequence.

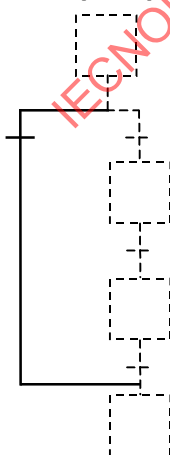
In this example, a priority is given to the transition 5/6, which is cleared when "a" is true.



EXAMPLE 3: Selection of sequences following synchronization of two preceding sequences.

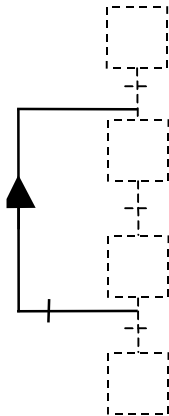
The selection of the succeeding sequences, by g and h, is possible only when the two transitions are cleared by the simultaneous activity of steps 8 and 9 (symbol 9).

6.2.4 Step skip



Particular case of selection of sequences, which allows a complete sequence or one or several steps of the sequence to be skipped, when, for example, the actions associated to these steps become unnecessary.

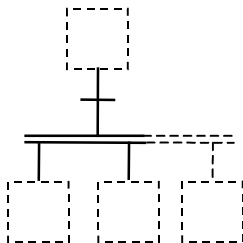
6.2.5 Backward sequence skip



Particular case of selection of sequences, which enables a sequence to be repeated until, for example, an established condition is satisfied.

NOTE It is possible, for graphical representation reasons, to place transitions on horizontal directed links (see Note 3 symbol 7).

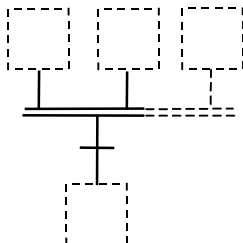
6.2.6 Activation of parallel sequences



The synchronisation symbol 9 is used in this structure to indicate the simultaneous activity of several sequences from one or several steps.

NOTE After their simultaneous activation, the evolution of the active steps in each of the parallel sequences thus becomes independent.

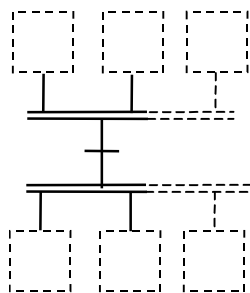
6.2.7 Synchronization of sequences



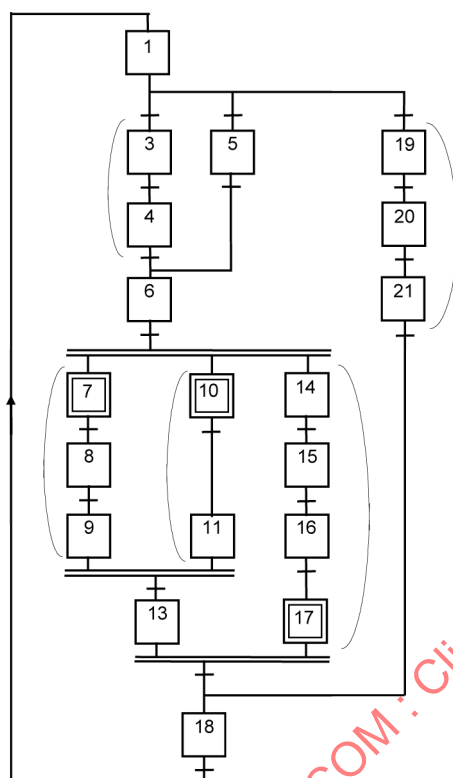
The synchronisation symbol 9 is used in this structure to indicate the delay before preceding sequences end before the activation of the succeeding sequence.

NOTE The transition is only enabled when all the preceding steps are active.

6.2.8 Synchronization and activation of parallel sequences



The synchronisation symbol 9 is used twice in this structure to indicate the delay before preceding sequences end before the simultaneous activation of the succeeding sequences.



EXAMPLE: Grafcet in which the following basic structures can be distinguished:

- the sequences (some of them are marked by parentheses),
- a selection of sequences (from step 1 to steps 3, 5 and 19),
- an activation of the parallel sequences (downstream from step 6),
- two synchronisations of sequences (from steps 9 and 11 to step 13, and from steps 13 and 17 to step 18).

NOTE 1 This example shows only the structure of the grafcet, its interpretation is not described.

NOTE 2 This grafcet is not a typical example because a grafcet is not necessarily looped back.

IECNORM.COM : Click to view the full PDF of IEC 60848:2013

6.3 Particular structures

6.3.1 Starting of a sequence by a source step



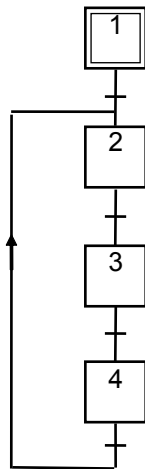
A source step is a step which does not have any preceding transition.

NOTE 1 To allow the activation of the source step, at least one of the following conditions, shall be satisfied:

the source step is initial,

the source step is required by a forcing order from a partial grafcet of the higher level (see 7.3),

the source step is one of the activated steps of an enclosure (see 7.4).

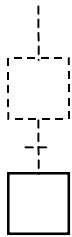


EXAMPLE 1: Initial source step:

The initial source step 1 is only active at the initialization time, the steps 2, 3, and 4 form a cycle of a single sequence.

NOTE 2 Only the grafcet structure is represented, its interpretation is not described.

6.3.2 End of a sequence by a pit step



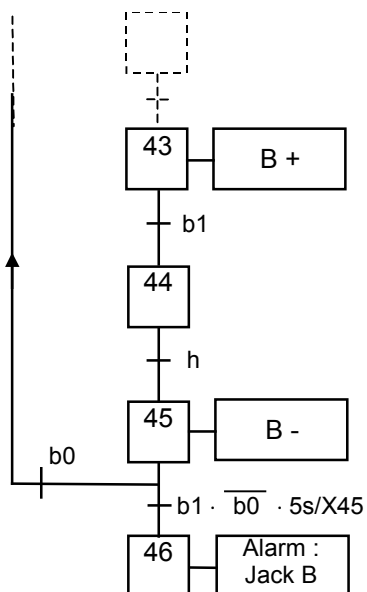
A pit step is a step which does not have any succeeding transition.

NOTE 1 The deactivation of the pit step is possible by only one of the two following ways:

a forcing order from a higher level partial grafcet (see 7.3),

the deactivation of the enclosing step if the pit step is enclosed there (see 7.4).

NOTE 2 A step may be source and pit at the same time, it then forms a single step sequence used to show a combinatorial behaviour.

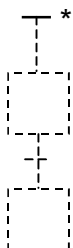


EXAMPLE: Pit step:

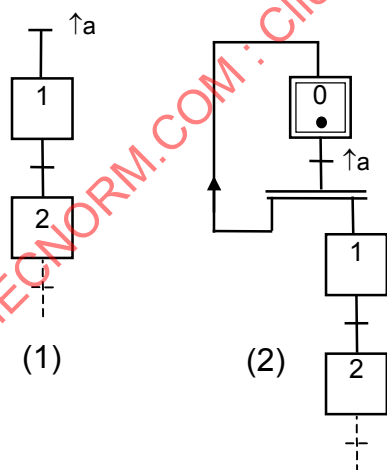
Pit step 46 is only activated if the logical condition " $b1 \cdot \overline{b0}$ " is verified 5 s after the activation of step 45 (see symbol 18). The output "Alarm: Jack B" is then assigned the true value.

6.3.3 Starting of a sequence with a source transition

A source transition is a transition, which does not have any preceding step. By convention, the source transition is always enabled and it is cleared as soon as its transition-condition * is true.



NOTE 1 The activation of the succeeding step of a source transition is effective as long as its transition-condition remains true, independent of the state of the transition-conditions for transitions enabled by this step (see evolution rule No. 5, 4.5.5). To avoid a continuous activation of the succeeding step of the source transition, it is better for the associated transition-condition to become true only when an input event or an internal event occurs. For that, the logical expression forming the transition-condition shall always include an input edge.

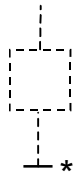


EXAMPLE: Source transition and equivalent structure:

Representations (1) and (2) describe an equivalent behaviour: step 1 is activated each time the Boolean variable a changes from value 0 to value 1. The representation (1) uses the source step, the representation (2) uses the synchronization symbol and a loop back to maintain initial step 0 active.

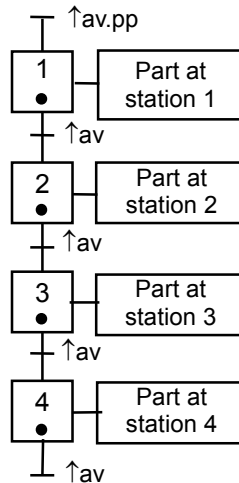
NOTE 2 The dot in step 0 indicates that this step is permanently active.

6.3.4 End of a sequence by a pit transition



A pit transition is a transition, which has no succeeding step.

NOTE 1 When the pit transition is enabled and when its associated transition-condition * is true, the only consequence of the clearing of the transition is the deactivation of the upstream steps.



EXAMPLE: structure of a shift register:

The structure of a shift register is a pertinent use of a source transition and of a pit transition. In this example, each active step indicates the presence of a part at the corresponding station. The presence of a part (pp) at the entry and the advance of the transfer between stations (↑av) activates step 1 by the clearing of the source transition. On each advance of the transfer (↑av) the enabled transitions are simultaneously cleared, including the pit transition downstream of the step 4.

NOTE 2 The representation corresponds to the frequent case when all the steps are simultaneously active.

7 Structuring

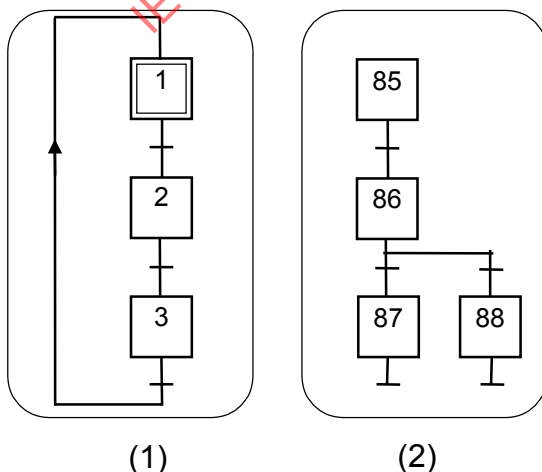
7.1 General

The complexity of the automated systems requires means for the structuring of the specification. This structuring assisted or not by suitable methodologies, can be limited simply to the division of the specification or can integrate hierarchical concepts of forcing or enclosure.

7.2 Partition of a grafcet chart

7.2.1 Connected grafcet chart

A connected grafcet chart is a structure in which there is always a continuity of links (alternation of steps and transitions) between any two elements, step or transition, in the grafcet chart.



EXAMPLE:

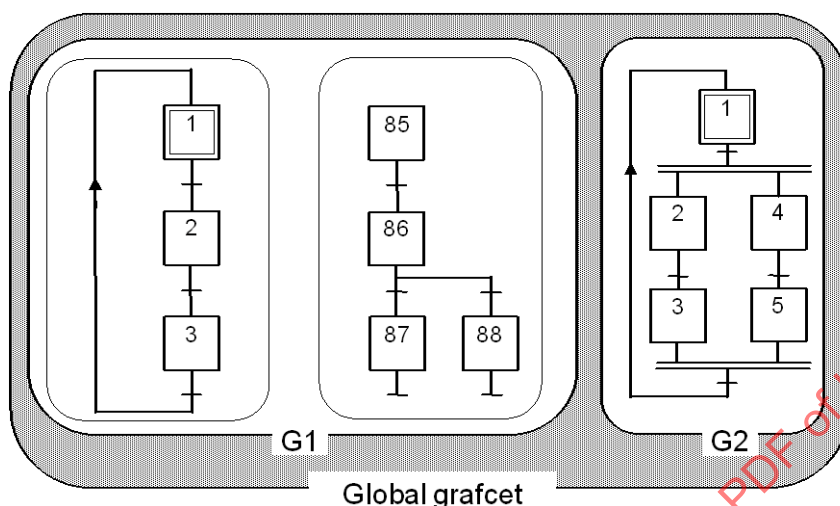
All the elements in frame (1) form a connected grafcet chart, since its steps and its transitions are connected by a directed link. The elements of the frame (2) also form a connected grafcet chart.

NOTE A non-connected grafcet has no technological meaning.

7.2.2 Partial grafcet

Formed by one or several connected grafcet charts, a partial grafcet chart (see Table 8) will results from the division of the global grafcet chart describing the behaviour of the sequential part of a system, according to the methodological criteria.

A partial grafcet chart is placed in a frame in which the label G^* of the represented grafcet chart is noted on the lower border.



EXAMPLE: Partition of a global grafcet:

The global grafcet is made of the partial grafcets G1 and G2. The partial grafcet G1 is made of two connected grafcets.

Table 8 – Partial grafcet chart

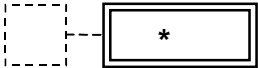
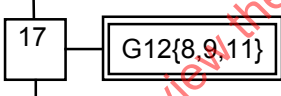
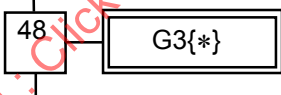
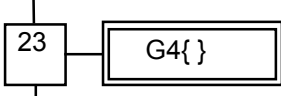
No.	Symbol	Description
[31]	G^*	Name of a partial grafcet: By convention, the letter G indicates a partial grafcet. The asterisk shall be replaced by the name of the partial grafcet.
[32]	XG^*	Partial grafcet variable: A partial grafcet is said to be active when at least one of its steps is active, consequently it is said to be inactive when none of its steps is active. The active or inactive state of a partial grafcet may be respectively represented by the logical values "1" or "0" of a variable XG^* in which the asterisk * is replaced by the name of the considered partial grafcet. The asterisk shall be replaced by the name of the partial grafcet. EXAMPLE: $XG1$ indicates the variable of the partial grafcet 1.
[33]	$G\#\{\dots\}$	Situation of a partial grafcet: The situation of a partial grafcet is represented by the set of its active steps at the considered moment. The situation of a partial grafcet # is given by $G\#\{\dots\}$ the contents of the curly brackets lists the active steps characterizing the situation of the partial grafcet at the considered moment. The sign # shall be replaced by the name of the partial grafcet. EXAMPLE: $G12\{8, 9, 11\}$ indicates the situation of the partial grafcet 12 at the considered moment, the situation in which only the steps 8, 9 and 11 are active.
[34]	$G\#\{*\}$	Current situation of a partial grafcet: The asterisk indicates, by default, the situation of the partial grafcet # at the considered moment. The sign # shall be replaced by the name of the partial grafcet.
[35]	$G\#\{\}$	Empty situation of a partial grafcet: Designates the situation of the partial grafcet # when none of its steps is active. The sign # shall be replaced by the name of the partial grafcet.
[36]	$G\#\{INIT\}$	Initial situation of a partial grafcet: Designates the situation of a partial grafcet # at the initial moment. The sign # shall be replaced by the name of the partial grafcet.

7.3 Structuring using the forcing of a partial grafcet chart

This means of structuring the specification of the sequential part of a system uses forcing orders (see Table 9). These orders allow the imposition of a specific situation to a given partial grafcet chart, from the situation of another (see example, Annex B). A partial grafcet chart may be forced by different preceding steps and orders.

NOTE "Orders are similar to continuous actions. While an action influences a variable, an order influences a partial grafcet chart."

Table 9 – Forcing of a partial grafcet chart

No.	Symbol	Description
[37]		Forcing order of a partial grafcet: Symbol in which the asterisk shall be replaced by a situation of a partial grafcet (symbols 33 to 36). Associated with the activity of a step of a hierarchically higher partial grafcet, the forcing order is an internal order which allows the imposition of a situation on a hierarchically lower partial grafcet. The forcing order is represented in a double rectangle associated with the step, to distinguish it from an action. The forcing order is an internal order for which the execution has priority on the application of the evolution rules. The forced grafcet cannot evolve during the period of the forcing order. The grafcet is said to be frozen. The use of forcing orders in a specification requires a hierarchical structure using partial grafcets in such a way that every forcing partial grafcet is at a higher level than the partial forced grafcets.     EXAMPLE 1: Forcing of a partial grafcet to an explicit situation. When step 17 is active, the partial grafcet 12 is forced to the situation characterised by the activity of the steps 8, 9 and 11. EXAMPLE 2: Forcing of a partial grafcet to the current situation. When step 48 is active, the partial grafcet 3 is forced to the situation in which it is at the forcing time. NOTE 1 This order is also named "freezing order". EXAMPLE 3: Forcing of a partial grafcet to the empty situation. When step 23 is active, the partial grafcet 4 is forced to the empty situation (see symbol 35). NOTE 2 In this case, no step of G4 is active. EXAMPLE 4: Forcing of a partial grafcet to the initial situation. When step 63 is active, the partial grafcet 8 is forced to the situation in which only its initial steps are active.

7.4 Structuring using the enclosure

A set of steps (partial grafcet) are said to be enclosed by a step, referred to as the enclosing step (see Table 10), if and only if, when this enclosing step is active, at least one of the enclosed steps is active. The designer can use the enclosure to structure a grafcet chart hierarchically (see example in Figure 3 or Figure B6). An enclosure can only belong to one enclosing step.

Table 10 – Enclosing steps

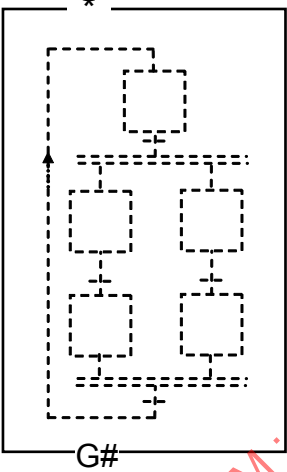
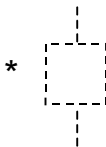
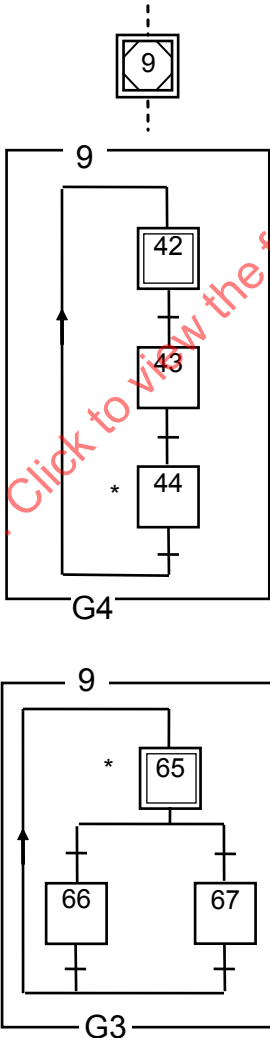
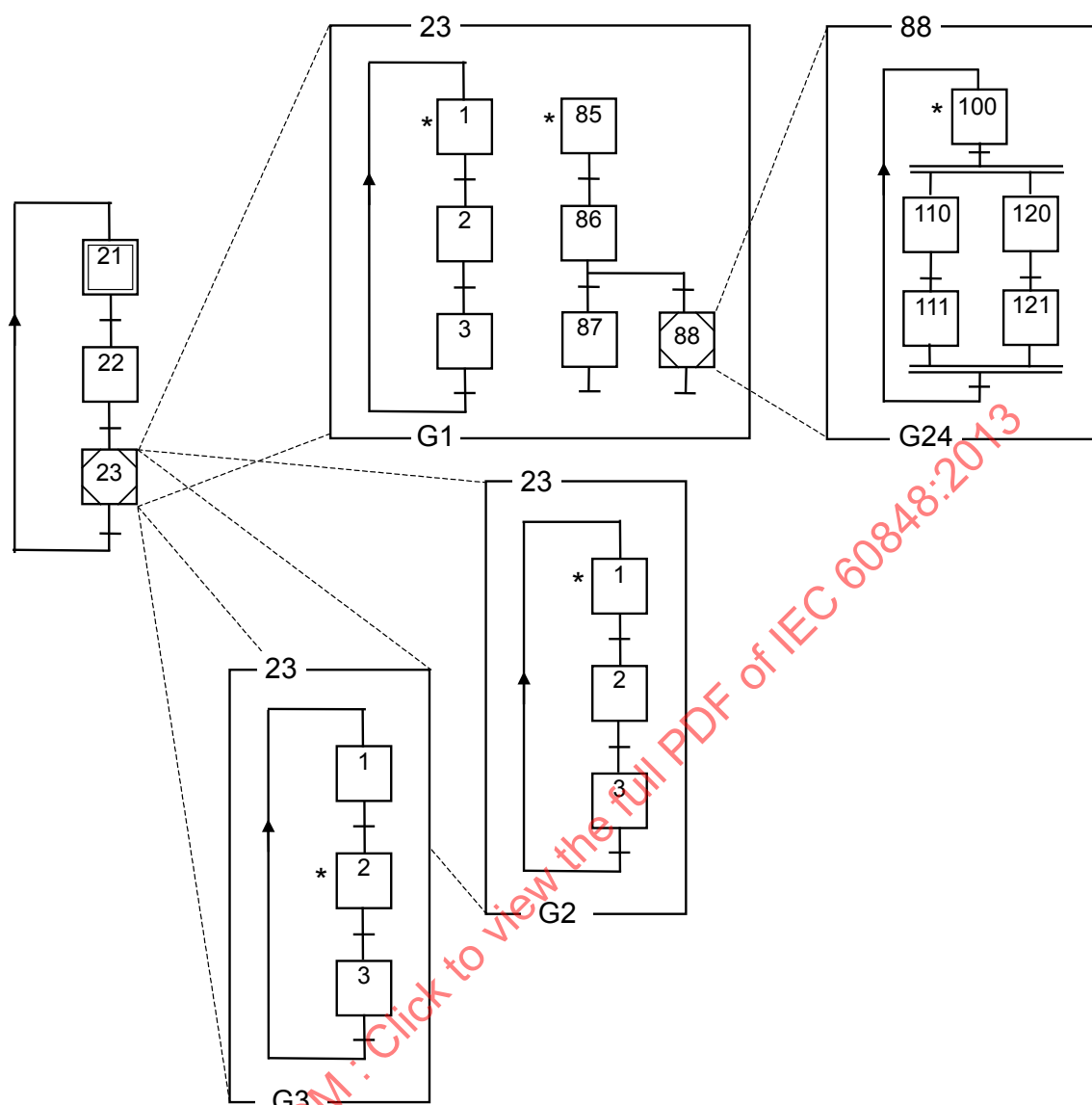
No.	Symbol	Description
[4]		Enclosing step (reminder of symbol 4): This notation indicates that this step contains other steps, referred to as enclosed, in one or several enclosures of the same enclosing step. The enclosing step has all the properties of the step, the step label shall replace the asterisk. The enclosing step may lead to one or several enclosures having at least one active step when the enclosing step is active in each of them, and having no active step when the enclosing step is inactive.
[38]		Graphic representation of an enclosure: An enclosure # of an enclosing step * may be represented by the partial grafcet of the enclosed steps grouped in a frame in which the name * of the enclosing step is placed at the top left side and the label G# of the represented enclosure is placed on the lower left side. In an enclosure, the set of the enclosed steps shall constitute a partial grafcet chart whose name may be used as a label for the corresponding enclosure.
[39]	$X^*/G\#$	Global designation of an enclosure: An enclosure # of an enclosing step * may be globally described by a textual expression in which the enclosing step * is designated by the step variable X^*, the enclosure by the symbol /, and the enclosed steps by the name of the partial grafcet G# to which they belong. NOTE This representation supposes that the designated partial grafcet was previously defined.
[40]	$X^*/X\#$	Elementary designation of an enclosure: A textual expression can indicate that a step # is enclosed in an enclosing step * by using the step variables, without naming the enclosure. NOTE This notation suits the design of a hierarchical series of steps enclosed in one another. It also allows a relative identification of the steps by level of enclosure. EXAMPLE: $X_4/X_{25}/X_{12}$ designates the enclosure of step 12 in step 25, which is itself enclosed in step 4.

Table 10 (continued)

No.	Symbol	Description
[5]		Initial enclosing step (reminder of symbol 5): This representation indicates that this step participates in the initial situation. In that case, at least one of the steps enclosed in each of its enclosures shall also be an initial step.
[41]		Activation link, general symbol. Represented by an asterisk at the left of the symbols of the enclosed steps, the activation link indicates the enclosed steps which are active at the activation of the enclosing step. The activation link should not be confused with the symbol for the initial steps, which may be enclosed. However, it is possible that an enclosed initial step also has an activation link. The deactivation of an enclosing step leads to the deactivation of all its enclosed steps. This deactivation often occurs at the clearing of the succeeding transition of the enclosing step but may also result from any other means of deactivation (forcing or enclosing of a higher level).  EXAMPLE: The enclosing step 9 is necessarily an initial step, because it encloses the initial step 42. The enclosure G4, of the enclosing step 9, contains steps 42, 43 and 44. The initial step 42 participates in the initial situation, it is therefore active at the initial time. On the other hand and for every activation of the step 9, and following of the grafcet evolution, step 44 is activated. The enclosure G3, of the enclosing step 9, contains steps 65, 66 and 67. The initial step 65 participates in the initial situation, it is therefore active at the initial time. It is also activated for every activation of step 9, as a consequence of the grafcet evolution.



IEC 367/13

EXAMPLE: Structuring by enclosure:

The enclosing step 23 has three enclosures represented by the partial grafcet G1, G2 and G3. The partial grafcet 24 is enclosed in step 88 of the partial grafcet 1. When the enclosing step 23 is activated, the steps 1 and 85 of G1 are also activated (the same is true for the other enclosing steps of 23: G2 and G3). Thus the partial grafcets G1, G2 and G3 are executed in parallel.

When the enclosing step 88 is activated, step 100 of G24 is also activated.

The deactivation of step 88 deactivates all steps of G24.

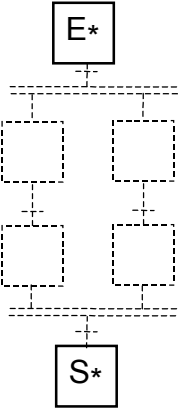
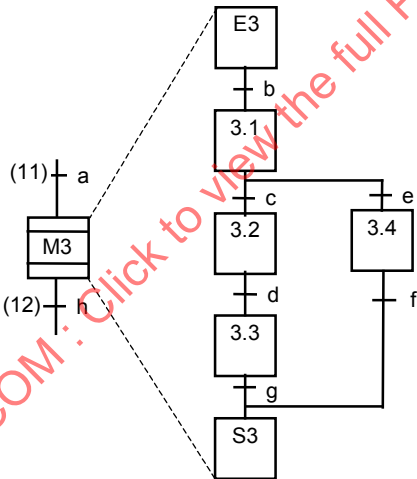
The deactivation of step 23 deactivates all the steps of G1, G2, G3 and, if step 88 was active, all the steps of G24.

Figure 3 – Example of grafcet with enclosures (including description)

7.5 Structuring using the macro-steps

To improve the understanding of grafcet charts, the specifications can be represented on several levels by "macro-representation" depicting the function to be performed without worrying about all the details, which are superfluous at the actual state of the description. The use of macro-steps (see Table 11) allows a gradual description from the general to the particular.

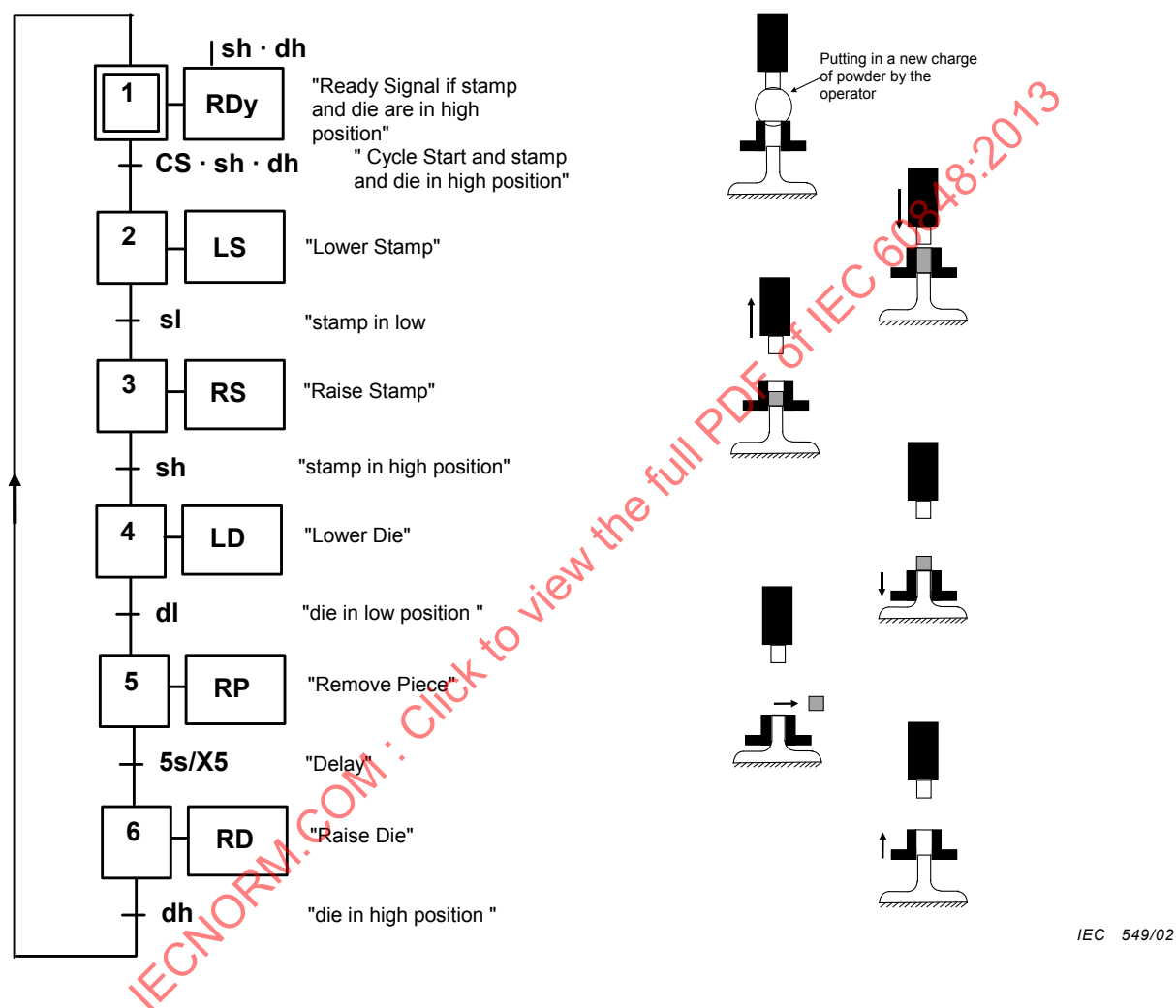
Table 11 – Macro-steps

No.	Symbol	Description
[6]		Macro-step (reminder of symbol 6): Unique representation of a detailed part of the grafset referred to as the expansion of the macro-step. An expansion can only belong to just one macro step. The macro-step does not have all the properties of the other kinds of step (symbols 1 to 5) because only its exit step (see symbol 42) can validate its succeeding transitions. A macro-step label shall replace the asterisk.
[42]		Expansion of the macro-step: The expansion of a macro-step M* is a part of the grafset with one entry step E* and one exit step S*. The entry step E* becomes active when one of the preceding transitions of the macro-step is cleared. One or all-succeeding transitions of the macro-step can be enabled only when the exit step S* is active. NOTE 1 The expansion of a macro-step can consist of one or several initial steps. NOTE 2 The expansion of a macro-step can consist of one or several macro-steps. EXAMPLE: Macro-step M3 represented with its expansion: The clearing of the transition 11 leads to the activity of the entry step E3 of the macro-step M3. Transition 12 is enabled only when exit step S3 is active. The clearing of transition 12 leads to the inactivity of step S3. 
[43]		Macro-step variable: A macro-step is said to be active when, at least, one of its steps is active, consequently it is said to be inactive when none of its steps is active. The active or inactive state of a macro-step can be represented by the logical values "1" or "0" respectively of a variable XM* in which the asterisk * shall be replaced by the name of the considered macro-step.

Annex A (informative)

Example of the control of a press

A press for compressed powders works as shown on the grafcet chart of Figure A.1. When the press is in stand-by at step 1, the stamp and the die are in high position and a "ready" signal is lighted to indicate to the operator to put in a new charge. The actions are successively performed as shown on the grafcet chart.



Codes:

Inputs

CS	Cycle Start
sh	stamp in high position
sl	stamp in low position
dh	die in high position
dl	die in low position

Outputs

RDy	Ready signal
LS	Lower Stamp
RS	Raise Stamp
LD	Lower Die
RP	Remove Piece
RD	Raise Die

Figure A.1 – Representation of the working press using a grafcet

Annex B (informative)

Example: Automatic weighing-mixing

B.1 Presentation of the example

Products A and B, previously weighed on a weighing unit C, and soluble bricks, brought one by one on a belt, are fed into a mixer N. The automatic system described in Figure B.1 allows a mixture of these three components to be obtained.

B.2 Cycle

Actuating the push-button "cycle start" causes the simultaneous weighing of products and the transport of bricks as follows:

- weighing product A up to the mark "a" of the weighing unit, and then dosing product B up to the mark "b" followed by emptying weighing unit C into the mixer;
- transport of two bricks.

The cycle ends with the mixer rotation and its final tipping after time "t1". The rotation of the mixer continues during emptying.

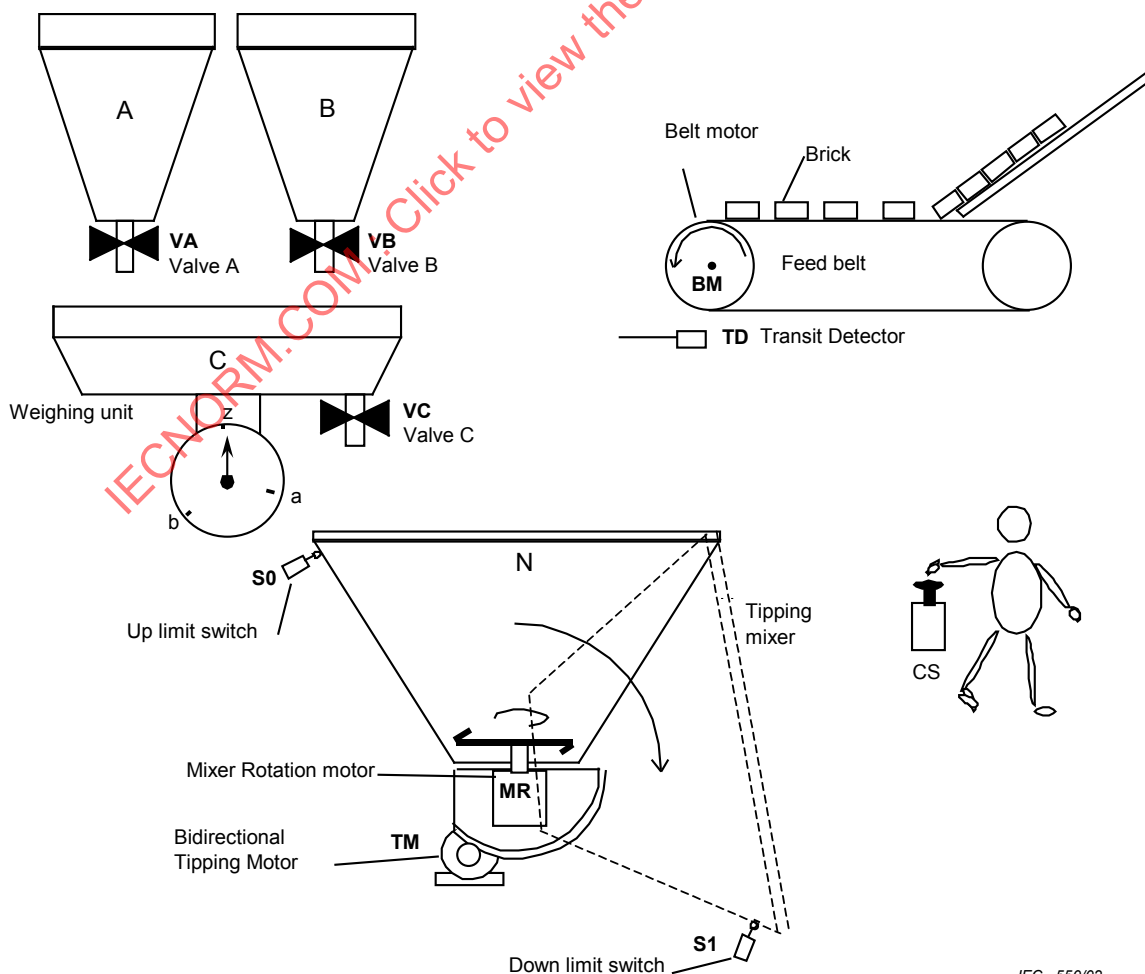
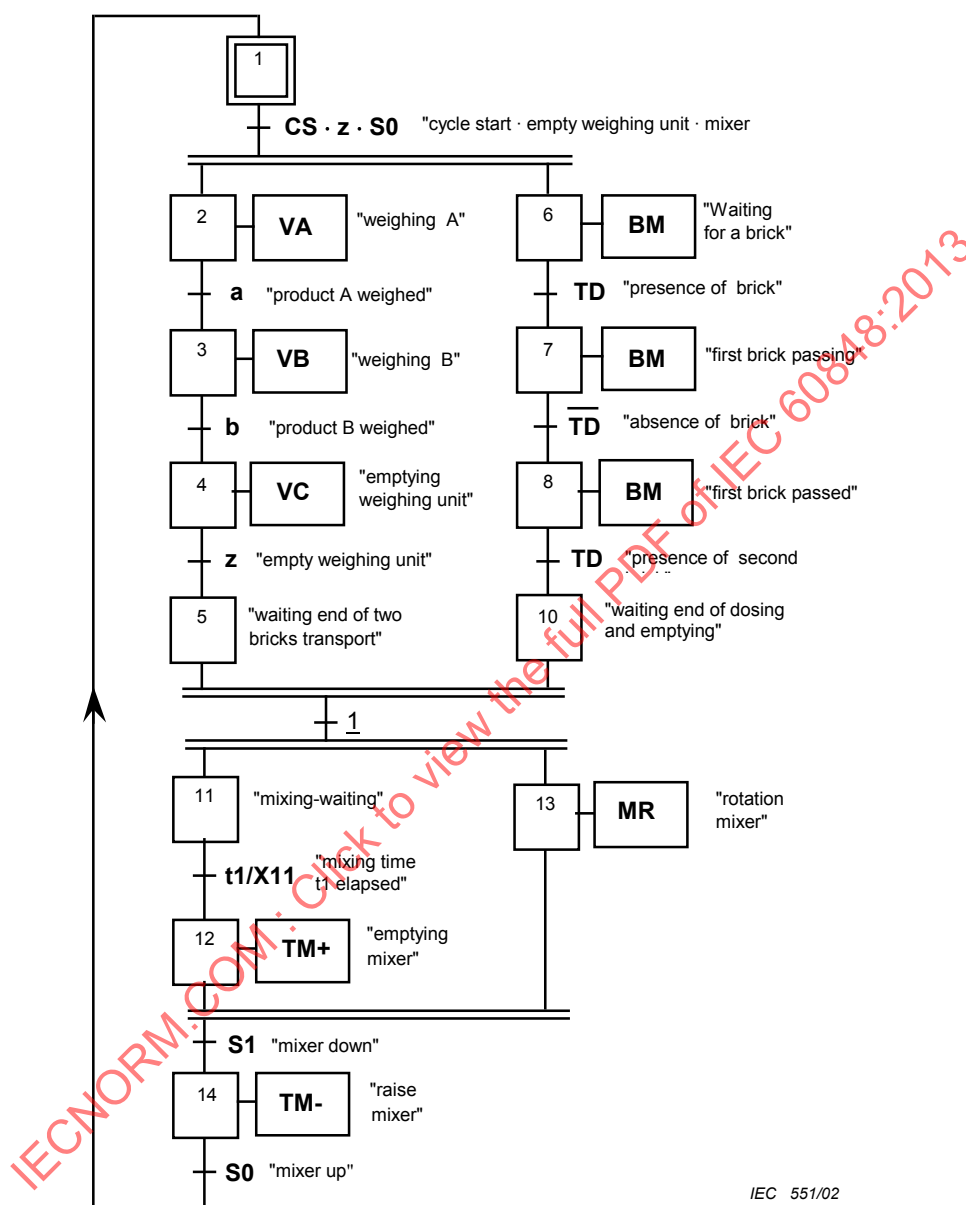


Figure B.1 – Overview diagram of weighing-mixing system

B.3 Behaviour description of the control command of the weighing-mixing

The logical behaviour of the weighing-mixing control command can be described by any of the grafset charts of Figures B.2 to B.4.



Codes:

Inputs

Outputs

CS	Cycle Start	BM	Belt Motor
TD	Transit Detector	MR	Mixer Rotation motor
a	Fluid weight A reached	TM+	Tipping Motor (down)
b	Fluid weight A + B reached	TM-	Tipping Motor (up)
z	Empty weighing unit	VA	Opening Valve A
S0	Mixer up	VB	Opening Valve B
S1	Mixer down	VC	Opening Valve C

Figure B.2 – Grafset of a weighing-mixing involving only continuous actions

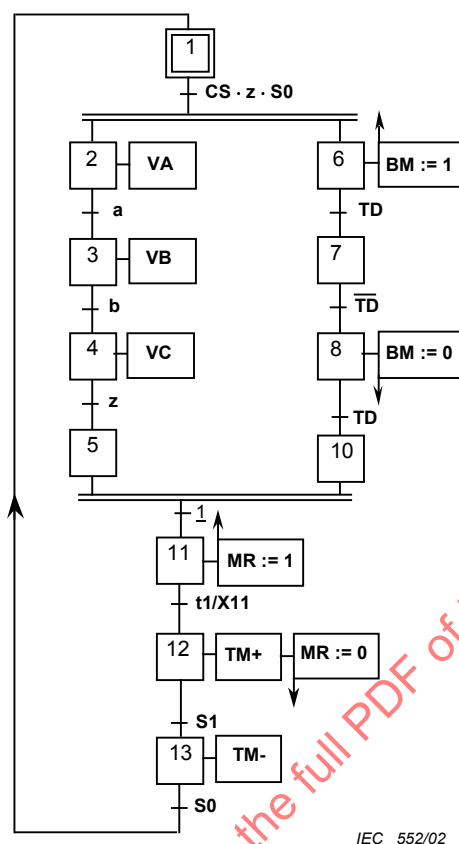


Figure B.3 – Grafcet of the weighing-mixing, involving continuous and stored actions

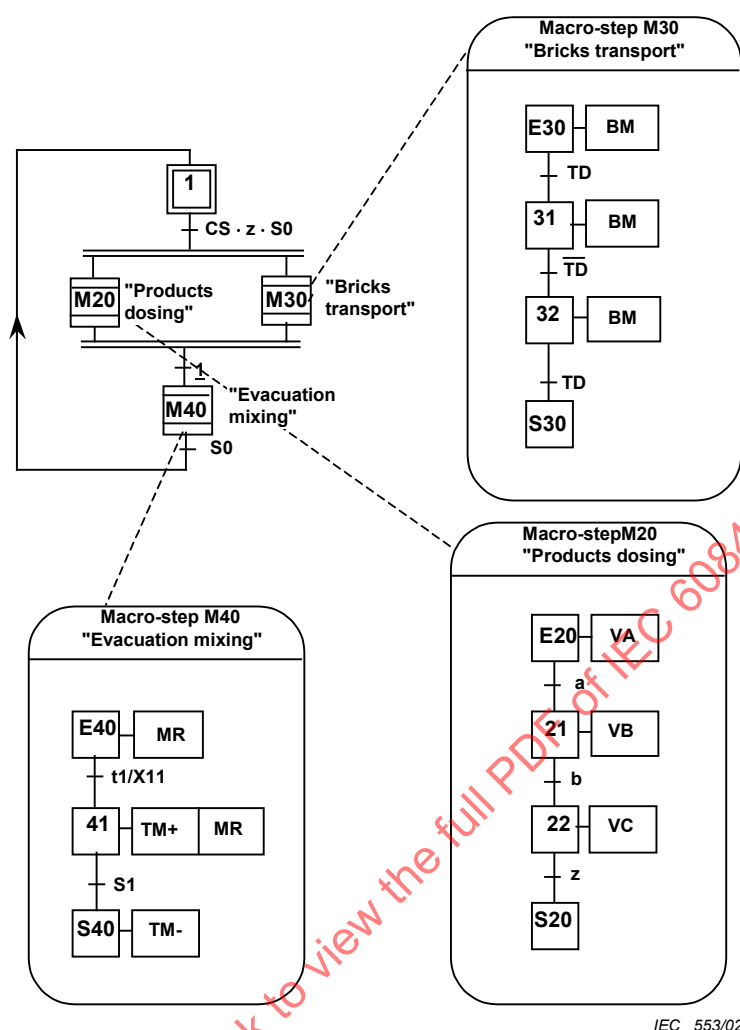
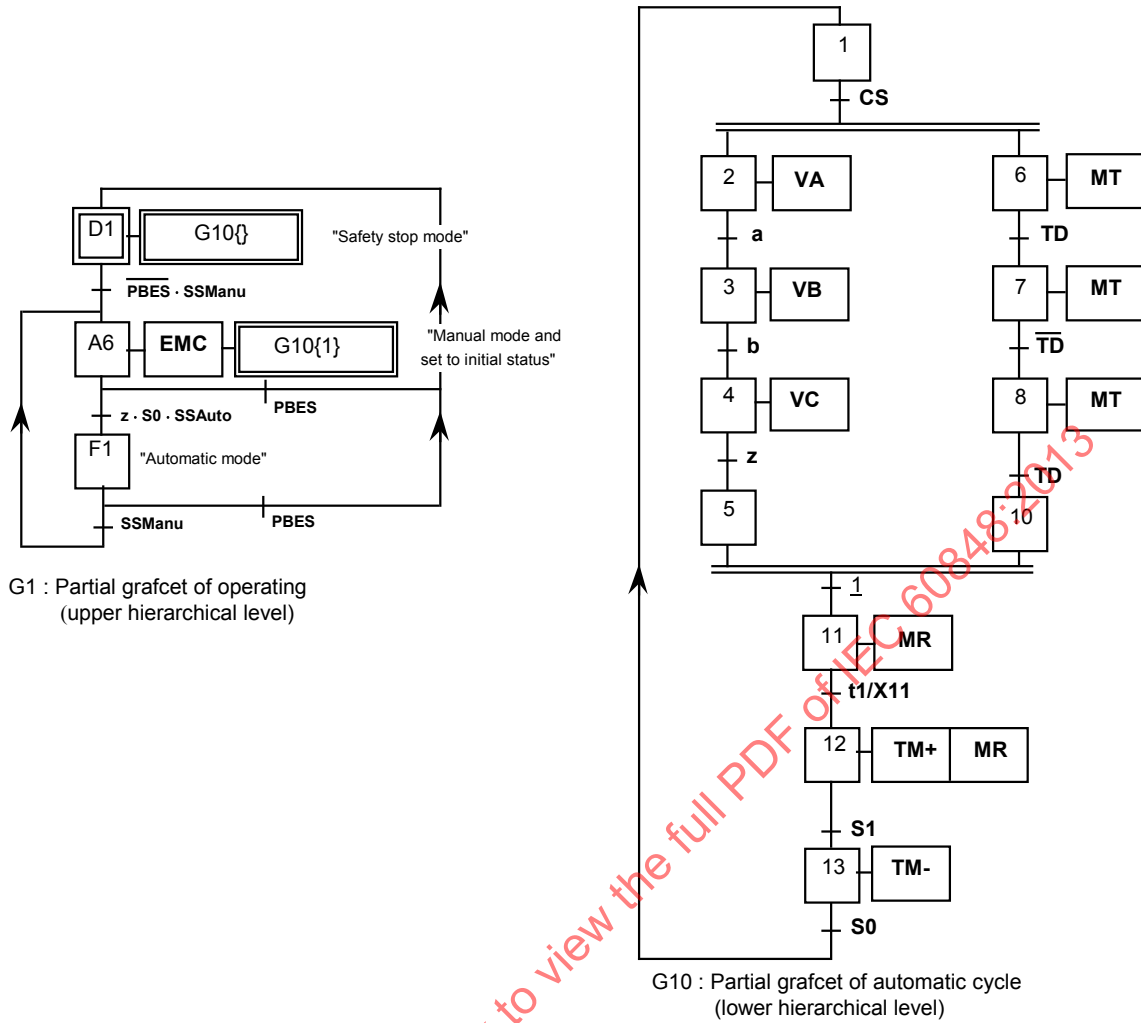


Figure B.4 – Grafset of the weighing-mixing, divided into a global description using macro-steps and a description detailed by the macro-step expansions

B.4 Weighing-mixing: structuring according to operating modes

Taking into account the operating modes of the automatic weighing-mixing can lead to the hierarchical structuring of the specification by using forcing orders (Figure B.5) or enclosing steps (Figure B.6). The complementary inputs and outputs given in the table in Figure B.5 are necessary to take into account the orders from the operator.



IEC 554/02

Codes:

Inputs

PBES	Push-button emergency stop
SSAuto	Selector-switch on auto mode
SSManu	Selector-switch on manual mode

Outputs

EMC	Enabled manual controls

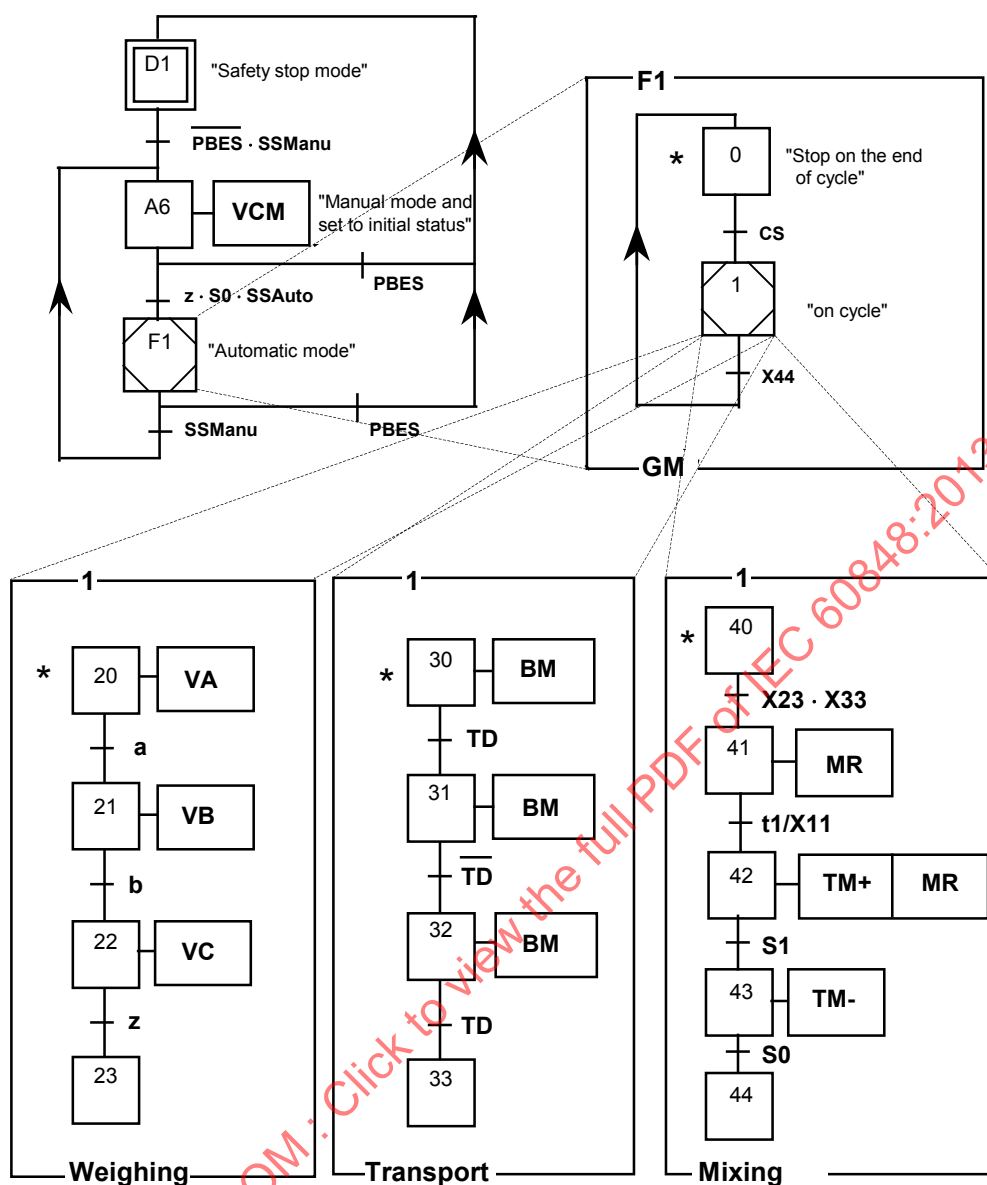
The forcing hierarchy involves two levels.

The forcing order, associated with the step D1 of the partial grafcet 1, forces the partial grafcet 10 to the empty situation (none of the steps of G10 is active).

The forcing order, associated with the step A6 of G1, forces G10 to the situation in which step 1 alone is active (but no transition is clearable).

The activation of the step F1 allows the normal evolution of G10 (because it does not depend of a forcing order).

Figure B.5 – Structuring with operating modes using forcing orders



IEC 555/02

The enclosure hierarchy involves three levels.

When step D1 is active, no other step is active.

When step A6 is active, no other step is active and the manual controls (EMC) are enabled.

When the enclosing step F1 is activated, step 0 of its enclosure GM is also activated.

When the enclosing step 1 is activated, step 20 of its "weighing" enclosure, step 30 of its "transport" enclosure and step 40 of its "mixing" enclosure are also activated.

Figure B.6 – Structuring with operating modes using enclosing step

Annex C (informative)

Relations between GRAFCET of IEC 60848 and the SFC of IEC 61131-3

C.1 Introduction

IEC 60848 and IEC 61131-3 each have a specific domain of application:

- a behaviour specification language (GRAFCET – GRAPhe Fonctionnel de Commande Etape Transition) independent of any specific technology of implementation, for IEC 60848, and
- a specific programming language (SFC – Sequential Function Chart), for IEC 61131-3.

GRAFCET of IEC 60848 is used by a grafcet chart to describe/specify the behaviour of system, as viewed from "outside" of the system, while the SFC language of IEC 61131-3 is used to describe (part of) the implemented software structure "inside" of the system.

If the two languages were both used to describe a control system, the two descriptions (two different document kinds) would in a given case look graphically similar. However, they would not have the same meaning, not even if they were graphically identical. This would just indicate that the structure of the software program, described in a software diagram, behaves in a way such that it can be described with a graphically similar grafcet chart. The properties of the underlying elements associated with the graphical element representations are nevertheless different in the two cases.

A specification using IEC 60848:2012 needs to be interpreted before implementation as a program using IEC 61131-3. There is presently no textual representation available for IEC 60848:2012 to support the interpretation and possible conversion into a program for an automation system.

C.2 In detail

- a) GRAFCET designates a language of specification of the logical behaviour of systems. This specification is independent of the technology of realisation considered.
- b) The SFC designates one of the programming languages defined in IEC 61131-3:1993. This language is inspired by IEC 60848:1988, but there is no identity between the two graphic representations and the semantics of the two languages.
- c) At the present time, no method is known to translate a GRAFCET specification language into a SFC program: it is necessary to translate the theoretical semantics of the first in the implemented semantics of the other. That is why, whereas more synthetic and more ergonomic, GRAFCET is generally used for a global specification and the SFC for the detailed conception.
- d) The revision of IEC 60848:1988, while preserving the existing syntax from the first edition and independence vis-à-vis the realisations, has been improved on the following points, allowing:
 - a better definition of elements and rules of evolution (internal events, distinction between assignation and allocation, transient evolution, etc.);
 - a finer interpretation of the behaviour specified, resorting to a principle or a realisation algorithm to understand the evolution in the difficult cases is therefore not required;
 - the emergence of help tools for the conception, capable of validating the specification notably by the calculation of property proofs;
 - the emergence of help tools for the realisation, capable of a guaranteed translation in diagrams or languages adapted to the technologies chosen;

- a better definition of structuring means (macro-steps, enclosing steps, etc.) associated to the GRAFCET language, capable of supporting the use of effective conception methods;
 - a better formalisation of interfaces (predicate, allocation) brings in the Boolean variables manipulated by GRAFCET and the other variable types which are necessary to complete the description of the behaviour of the target system.
- e) The search for a systematic identity between a GRAFCET element and its SFC corresponding element leads to the imposition of the programmed semantics of the second language into the first one. This approach would limit the role of the GRAFCET standard, IEC 60848:1988, solely to a general definition of symbols and rules intended to sustain definitions of the SFC elements of IEC 61131-3:1993.
- f) IEC 61131-3:1993 specifies programming languages especially adapted to the PLCs (Programmable Logic Controllers). That is why it seems unlikely that this programmed description of the behaviour can apply in other technological contexts (electromechanical, electronic, pneumatic, etc.).
- g) The evolution of distributed automatic devices, pre-wired and/or pre-programmed, creates the need to describe, in a formal and ergonomic way, the behaviour (guaranteed by the manufacturer) of these new devices in a language facilitating their integration in an automated application.

C.3 Possible future evolution

Facing other non-standardized candidates (for example the state chart), the GRAFCET notation benefits from an experience acquired by a big number of designers of automatic machines. This advantage should be used through an evolution of the standard to widen the domain of this language and to make it more formal and more structured.

A future integrated approach has been discussed by IEC subcommittee 3B and IEC Technical Committee 65 allowing a textual output of IEC 60848:1988 to be automatically converted into a structure and a program following IEC 61131-3:1993 and IEC 61499 respectively.

Bibliography

IEC 60617, *Graphical symbols for diagrams*. Available from <http://std.iec.ch/iec60617>

IEC 61131-3:2003, *Programmable controllers – Part 3: Programming languages*

ISO 5807:1985, *Information processing – Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts*

IECNORM.COM : Click to view the full PDF of IEC 60848:2013

IECNORM.COM : Click to view the full PDF of IEC 60848:2013

SOMMAIRE

AVANT-PROPOS	59
INTRODUCTION	61
1 Domaine d'application	62
2 Références normatives	62
3 Termes et définitions	62
3.1 Termes du GRAFCET	62
3.2 Termes, usage général	64
4 Principes généraux	64
4.1 Contexte	64
4.2 Le GRAFCET, un langage qui spécifie le comportement	65
4.3 GRAFCET, présentation sommaire	66
4.3.1 Généralités	66
4.3.2 La structure	66
4.3.3 L'interprétation	66
4.4 Règle de syntaxe	67
4.5 Règles d'évolution	68
4.5.1 Généralités	68
4.5.2 Situation initiale	68
4.5.3 Franchissement d'une transition	68
4.5.4 Évolution des étapes actives	68
4.5.5 Évolutions simultanées	68
4.5.6 Activation et désactivation simultanées d'une étape	68
4.6 Événements d'entrée	68
4.6.1 Généralités	68
4.6.2 Spécification des événements d'entrée	69
4.7 Événements internes	69
4.7.1 Généralités	69
4.7.2 Événements internes décrits par l'activation d'une étape	69
4.7.3 Événements internes décrits par la désactivation d'une étape	69
4.7.4 Événements internes décrits par le franchissement d'une transition	70
4.8 Modes de sortie	70
4.8.1 Généralités	70
4.8.2 Mode continu (assignation sur état)	70
4.8.3 Mode mémorisé (affectation sur événement)	70
4.9 Application des règles d'évolution	71
4.9.1 Généralités	71
4.9.2 Évolution non fugace	71
4.9.3 Évolution fugace	71
4.9.4 Conséquence d'une évolution fugace sur les assignations	72
4.9.5 Conséquence d'une évolution fugace sur les affectations	72
4.10 Comparaison entre les deux modes de sortie	73
4.10.1 Généralités	73
4.10.2 Détermination de la valeur des sorties	73
4.10.3 Analyse de la valeur des sorties d'un diagramme grafcet à un instant déterminé	73
4.10.4 Actions relatives à une évolution fugace	73
4.10.5 Conflit éventuel sur la valeur des sorties	74

5	Représentation graphique des éléments.....	74
6	Représentation graphique des structures de séquences	87
6.1	Généralités.....	87
6.2	Structures de base	87
6.2.1	Séquence	87
6.2.2	Cycle d'une seule séquence.....	88
6.2.3	Sélection de séquences	88
6.2.4	Saut d'étapes.....	89
6.2.5	Reprise de séquence	89
6.2.6	Activation de séquences parallèles.....	89
6.2.7	Synchronisation de séquences	89
6.2.8	Synchronisation et activation de séquences parallèles.....	90
6.3	Structures particulières	91
6.3.1	Début de séquence par une étape source	91
6.3.2	Fin de séquence par une étape puits.....	91
6.3.3	Début de séquence par une transition source	92
6.3.4	Fin de séquence par une transition puits	93
7	Structuration.....	93
7.1	Généralités.....	93
7.2	Partition d'un diagramme grafcet	93
7.2.1	Diagramme grafcet connexe	93
7.2.2	Diagramme grafcet partiel.....	94
7.3	Structuration par forçage d'un diagramme grafcet partiel.....	95
7.4	Structuration par encapsulation.....	96
7.5	Structuration par macro-étapes.....	98
	Annexe A (informative) Exemple de commande d'une presse	100
	Annexe B (informative) EXEMPLE: Doseur malaxeur automatique	101
	Annexe C (informative) Les relations entre le GRAFCET selon la CEI 60848 et le SFC selon la CEI 61131-3.....	107
	Bibliographie	109
	Figure 1 – Représentation graphique de la partie séquentielle d'un système	65
	Figure 2 – Éléments de structure et d'interprétation utilisés dans un grafcet pour décrire le comportement de la partie séquentielle d'un système défini par ses variables d'entrée et de sortie	67
	Figure 3 – Exemple de grafcet avec encapsulation (et sa description).....	98
	Figure A.1 – Représentation du fonctionnement de la presse par un grafcet	100
	Figure B.1 – Représentation schématique du doseur malaxeur	101
	Figure B.2 – Grafcet du doseur malaxeur, ne comportant que des actions continues.....	102
	Figure B.3 – Grafcet du doseur malaxeur comportant des actions continues et des actions mémorisées	103
	Figure B.4 – Grafcet du doseur malaxeur décomposé en une description globale utilisant des macro-étapes, et une description détaillée par les expansions de ces macro-étapes.....	104
	Figure B.5 – Structuration selon les modes de marche utilisant des ordres de forçage	105
	Figure B.6 – Structuration selon les modes de marche utilisant des étapes encapsulantes.....	106

Tableau 1 – Étapes	75
Tableau 2 – Transitions	76
Tableau 3 – Liaisons orientées	77
Tableau 4 – Conditions de transition associées	78
Tableau 5 – Actions continues	82
Tableau 6 – Actions mémorisées	85
Tableau 7 – Commentaires associés aux éléments d'un grafcet.....	87
Tableau 8 – Diagramme grafcet partiel	94
Tableau 9 – Forçage d'un diagramme grafcet partiel.....	95
Tableau 10 – Étapes encapsulantes	96
Tableau 11 – Macro-étapes	99

IECNORM.COM : Click to view the full PDF of IEC 60848:2013

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**LANGAGE DE SPÉCIFICATION GRAFCET POUR
DIAGRAMMES FONCTIONNELS EN SÉQUENCE****AVANT-PROPOS**

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de brevet. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale CEI 60848 a été établie par le sous-comité 3B: Documentation, du comité d'études 3 de la CEI: Structures d'informations, documentation et symboles graphiques.

Cette troisième édition annule et remplace la deuxième édition parue en 2002, dont elle constitue une révision technique générale comprenant la définition étendue du concept de variables qui introduit: la variable interne, la variable d'entrée et la variable de sortie.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
3/1135/FDIS	3/1138/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IECNORM.COM : Click to view the full PDF of IEC 60848:2013

INTRODUCTION

Cette Norme internationale est destinée principalement aux utilisateurs tels que les concepteurs, réalisateurs, agents de maintenance, etc. qui ont besoin de spécifier le comportement d'un système par exemple la commande d'un système d'automatisation, d'un composant de sûreté, etc. Ce langage de spécification peut également servir de moyen de communication entre les concepteurs et les utilisateurs de systèmes automatisés.

IECNORM.COM : Click to view the full PDF of IEC 60848:2013

LANGAGE DE SPÉCIFICATION GRAFCET POUR DIAGRAMMES FONCTIONNELS EN SÉQUENCE

1 Domaine d'application

La présente Norme internationale définit le langage de spécification GRAFCET¹ pour la description fonctionnelle du comportement de la partie séquentielle des systèmes de commande.

Cette norme définit les symboles et les règles nécessaires à la représentation graphique de ce langage, ainsi que l'interprétation qui en est faite.

Cette norme a été établie pour les systèmes automatisés de production des applications industrielles, cependant aucun champ d'application n'est exclu.

Les méthodes de réalisation d'une spécification utilisant le GRAFCET ne font pas partie du domaine d'application de cette norme. Une méthode possible est l'utilisation du langage "SFC" décrit dans la CEI 61131-3, qui définit un ensemble de langages de programmation destinés aux automates programmables.

NOTE Voir l'Annexe C pour de plus amples informations sur les relations entre la CEI 60848 et les langages de réalisation comme le SFC de la CEI 61131-3.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

(vide)

3 Termes et définitions

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

NOTE Les définitions des termes de 3.1 ne s'appliquent que dans le contexte du langage de spécification GRAFCET.

3.1 Termes du GRAFCET

3.1.1 action

élément du langage GRAFCET associé à une étape, l'action indique le comportement d'une variable de sortie ou de variables internes

3.1.2 liaison orientée

élément du langage GRAFCET, les liaisons orientées indiquent les voies d'évolution en reliant les étapes aux transitions et les transitions aux étapes

¹ GRAFCET: GRaphe Fonctionnel de Commande Etape Transition.

3.1.3**diagramme grafcet**

diagramme fonctionnel utilisant le langage de spécification GRAFCET

Note 1 à l'article: Le terme "diagramme grafcet" est, en raccourci, remplacé par "grafcet".

3.1.4**événement d'entrée**

événement caractérisé par le changement de valeur d'une ou plusieurs variables d'entrée de la partie séquentielle du système

3.1.5**événement interne**

événement caractérisé par un événement d'entrée associé à la situation de la partie séquentielle du système

3.1.6**interprétation**

partie du langage de spécification GRAFCET permettant d'établir la relation entre:

- les variables d'entrées et la structure, par les conditions de transition; et
- les variables de sorties et la structure, par les actions

3.1.7**situation**

désignation de l'état du système décrit par le langage de spécification grafcet et caractérisé par les étapes actives à un instant considéré

3.1.8**étape**

élément du langage GRAFCET utilisé pour définir la situation de la partie séquentielle d'un système

Note 1 à l'article: Une étape est soit active soit inactive.

Note 2 à l'article: L'ensemble des étapes actives représente la situation du système.

3.1.9**évolution fugace**

évolution caractérisée par le franchissement de plusieurs transitions successives à l'occurrence d'un unique événement d'entrée

3.1.10**transition**

élément du langage GRAFCET qui indique la possibilité d'évolution d'activité entre deux ou plusieurs étapes

Note 1 à l'article: Cette évolution possible s'accomplit par le franchissement de la transition.

3.1.11**condition de transition**

élément du langage GRAFCET associé à une transition exprimant le résultat d'une expression booléenne

Note 1 à l'article: Une condition de transition est soit vraie soit fausse.

3.1.12**variable**

grandeur définie par son nom et sa valeur booléenne ou numérique

3.1.13

variable d'entrée

variable qui peut influencer le comportement décrit par le diagramme grafcet

EXEMPLE Variable BOOLEENNE indiquant la violation d'une limite de température.

Note 1 à l'article: La variable peut appartenir à l'environnement ou à un autre élément du système.

3.1.14

variable de sortie

variable qui peut être influencée par le comportement décrit par le diagramme grafcet

EXEMPLE Valeur de consigne d'un automate PID.

Note 1 à l'article: La variable peut appartenir à l'environnement ou à un autre élément du système.

3.1.15

variable interne

variable utilisée à l'intérieur d'un diagramme grafcet et invisible pour les autres composants du système ou pour l'environnement

EXEMPLES Variable d'étape X^* (symbole 2.1 du Tableau 1), durée d'étape T^* (symbole 2.2 du Tableau 1), compteur de boucle dans un diagramme grafcet.

3.2 Termes, usage général

3.2.1

diagramme

représentation graphique décrivant le comportement d'un système, par exemple les relations entre deux ou plus de deux grandeurs variables, actions ou états

3.2.2

structure

partie du langage de spécification GRAFCET permettant de décrire l'évolution possible entre les situations

3.2.3

système

ensemble d'éléments reliés entre eux, considérés dans un contexte défini comme un tout et séparés de leur environnement

Note 1 à l'article: Les éléments du système peuvent être à la fois des objets matériels ou des concepts aussi bien que les résultats de ceux-ci (par exemple formes d'organisation, méthodes mathématiques, langages de programmation)

Note 2 à l'article: Le système est considéré comme séparé de l'environnement et des autres systèmes extérieurs par une surface imaginaire qui coupe les liaisons entre eux et le système.

Note 3 à l'article: Le langage GRAFCET peut être utilisé pour décrire le comportement logique de n'importe quel type de système.

[SOURCE: CEI 60050-351:1998, 351-11-01]

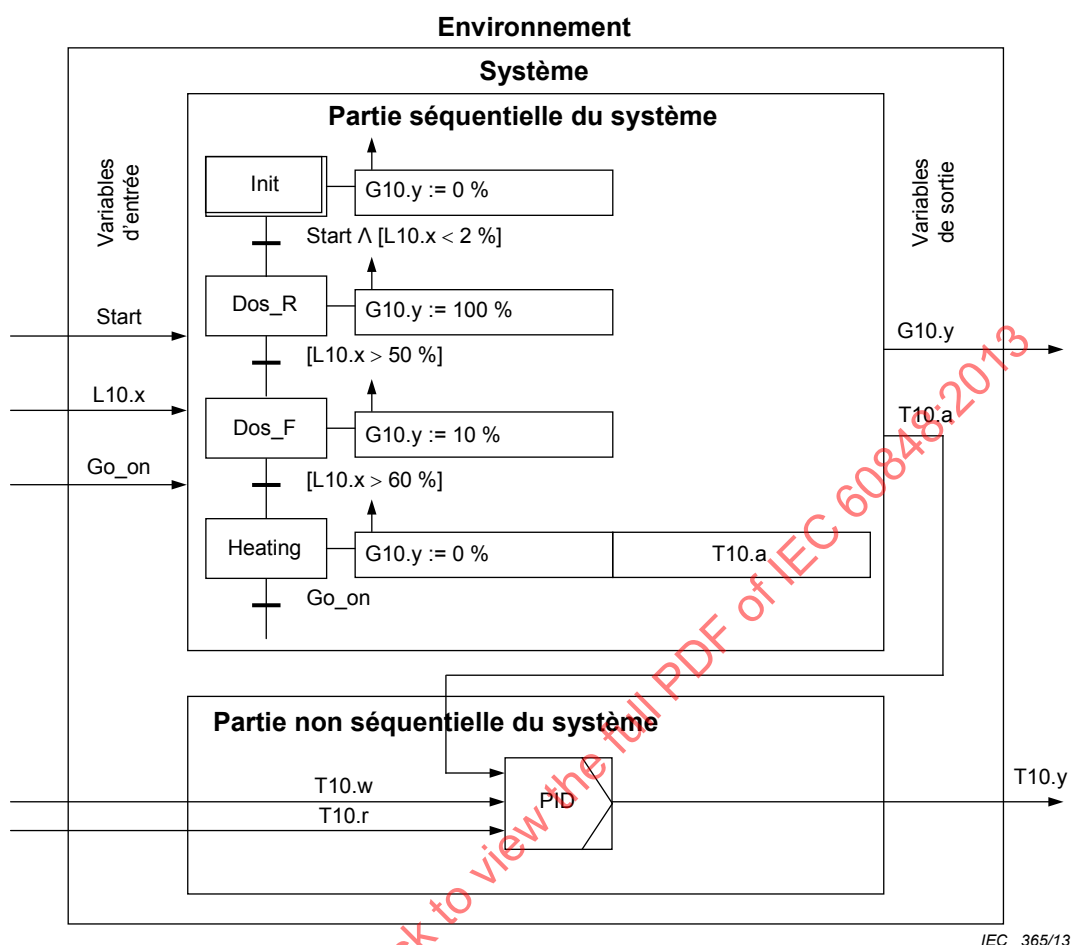
4 Principes généraux

4.1 Contexte

La réalisation d'un système automatisé requiert, notamment, une description liant les effets aux causes. Pour cela, on décrira l'aspect logique du comportement souhaité du système.

La partie séquentielle du système désigne l'aspect logique d'un système (voir Figure 1). Le comportement indique la manière dont les variables de sortie dépendent des variables

d'entrée. Le diagramme grafcet a pour objet de spécifier le comportement de la partie séquentielle des systèmes.



L10.x niveau du réservoir

G10.y vanne de dosage – position

T10.a boucle de température – mode automatique

T10.w boucle de température – consigne

T10.r boucle de température – valeur mesurée

T10.y boucle de température – valeur manipulée

Figure 1 – Représentation graphique de la partie séquentielle d'un système

4.2 Le GRAFCET, un langage qui spécifie le comportement

Le langage de spécification GRAFCET permet d'établir un diagramme grafcet exprimant le comportement attendu de la partie séquentielle d'un système déterminé. Ce langage se caractérise principalement par ses éléments graphiques qui, associés à une expression alphanumérique des variables, offre une représentation synthétique du comportement reposant sur une description indirecte de la situation du système.

La description du comportement sous forme d'états est la suivante: les états, "monomarqués", correspondent aux situations du diagramme de GRAFCET, ce qui implique l'unicité de la situation à un instant donné. Les états sont reliés les uns aux autres par une condition d'évolution, ce qui permet de décrire le passage d'une situation à une autre.

Pour des raisons de commodité, la description du comportement sous forme d'états est avantageusement remplacée par une description sous forme d'étapes appelée GRAFCET. Dans le diagramme de GRAFCET plusieurs étapes peuvent être actives simultanément, la situation étant alors caractérisée par l'ensemble des étapes actives à l'instant considéré. Les

conditions d'évolution d'un ensemble d'étapes vers un autre sont supportées par une ou plusieurs transitions, caractérisées chacune par:

- ses étapes amont,
- ses étapes aval,
- sa condition de transition associée.

NOTE La règle de syntaxe imposant l'alternance étape-transition résulte de ce qui précède.

4.3 GRAFCET, présentation sommaire

4.3.1 Généralités

Le langage de spécification GRAFCET est utilisé pour concevoir des grafjets donnant une représentation graphique et synthétique du comportement des systèmes. La représentation (voir Figure 2) distingue:

- la structure, qui permet de décrire les évolutions possibles entre les situations, et
- l'interprétation, qui permet la relation entre les variables d'entrées, la structure, et les variables de sorties (des règles d'évolution, d'assignation et d'affectation sont nécessaires pour réaliser cette interprétation).

Les symboles relatifs aux éléments GRAFCET, qui représentent les étapes d'un processus et les liens entre ses étapes, figurent avec des exemples dans les Tableaux 1 à 4 à l'Article 5.

4.3.2 La structure

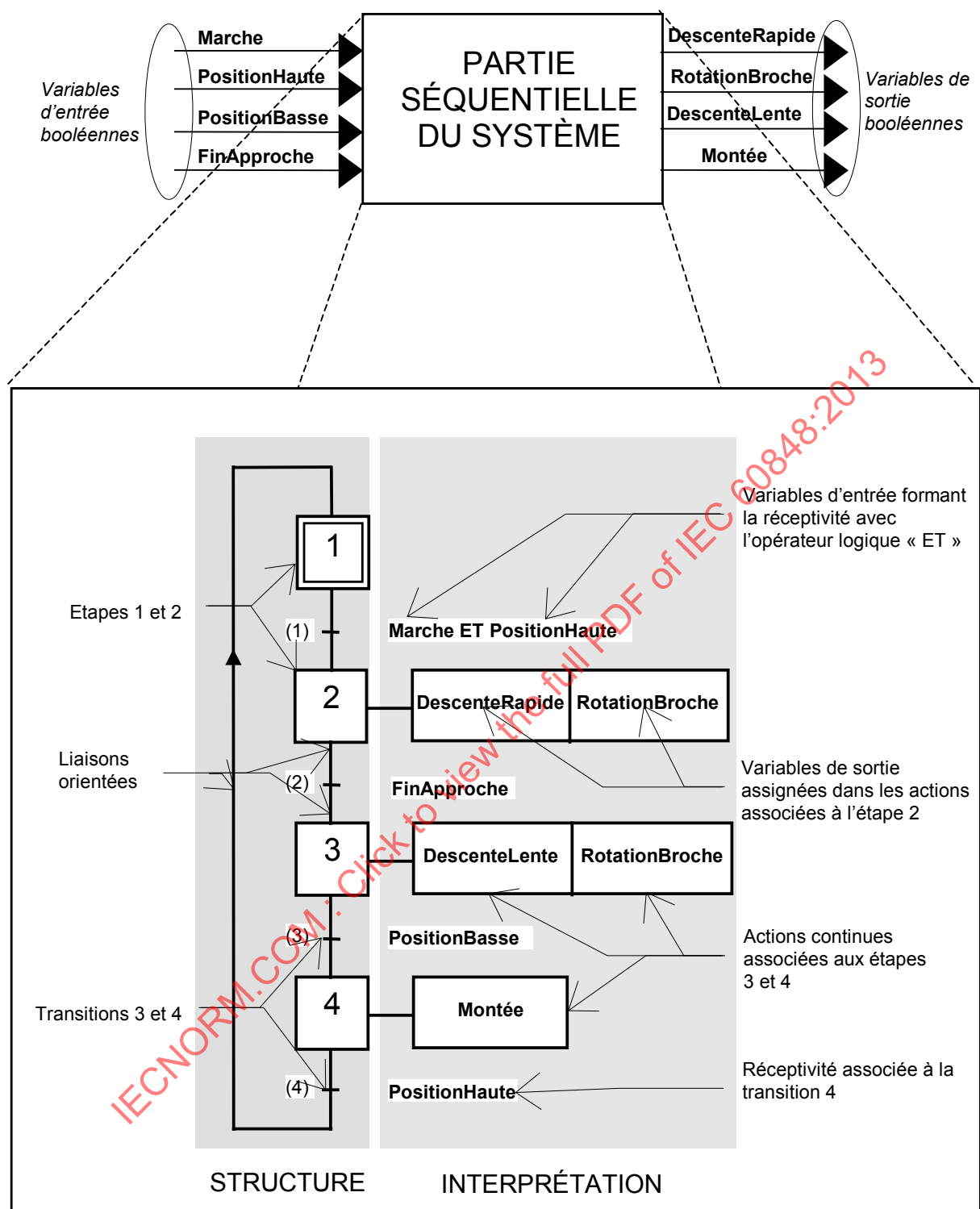
La structure est constituée des éléments de base suivants:

- Étape (définition: 3.1.8, symbole 1). Une étape est soit active, soit inactive, l'ensemble des étapes actives d'un grafjet à un instant donné représente la situation de ce diagramme grafjet à l'instant considéré.
- Transition (définition: 3.1.10, symbole 7). Une transition indique la possibilité d'évolution d'activité entre deux ou plusieurs étapes. Cette évolution s'accomplit par le franchissement de la transition.
- Liaison orientée (définition: 3.1.2, symbole 10). Une liaison orientée relie soit une ou plusieurs étapes à une transition, soit une transition à une ou plusieurs étapes.

4.3.3 L'interprétation

L'interprétation se fait grâce aux éléments suivants:

- Condition de transition (définition: 3.1.11, symbole 13). Associée à chaque transition, la condition de transition est une condition logique qui est soit vraie, soit fausse, et qui est composée de variables d'entrées et/ou de variables internes.
- Action (définition: 3.1.1). L'action indique, dans un rectangle, comment agir sur la variable de sortie ou interne, soit par assignation (action continue, symbole 20), soit par affectation (action mémorisée, symbole 26).



IEC 366/13

Figure 2 – Éléments de structure et d'interprétation utilisés dans un grafset pour décrire le comportement de la partie séquentielle d'un système défini par ses variables d'entrée et de sortie

4.4 Règle de syntaxe

L'alternance étape-transition et transition-étape doit toujours être respectée, quelle que soit la séquence parcourue.

Conséquences:

- Deux étapes ou deux transitions ne doivent jamais être reliées par une liaison orientée;
- La liaison orientée relie obligatoirement une étape à une transition ou une transition à une étape.

4.5 Règles d'évolution

4.5.1 Généralités

Sachant que toute situation est caractérisée par l'ensemble des étapes actives à l'instant considéré, les règles d'évolution ne sont que l'application, sur les étapes, du principe d'évolution entre les situations de la partie séquentielle du système.

4.5.2 Situation initiale

La situation initiale est la situation à l'instant initial, elle est donc décrite par l'ensemble des étapes actives à cet instant. Le choix de la situation à l'instant initial repose sur des considérations méthodologiques et relatives à la nature de la partie séquentielle du système visé.

Règle 1: La situation initiale, choisie par le concepteur, est la situation à l'instant initial.

4.5.3 Franchissement d'une transition

Règle 2: Une transition est dite validée lorsque toutes les étapes immédiatement précédentes reliées à cette transition sont actives. Le franchissement d'une transition se produit:

- lorsque la transition est validée, et
- lorsque la condition de transition associée à cette transition est vraie.

4.5.4 Évolution des étapes actives

Règle 3: Le franchissement d'une transition entraîne simultanément l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.

4.5.5 Évolutions simultanées

L'évolution entre deux situations actives implique qu'aucune situation intermédiaire ne soit possible et on passe donc instantanément par un ensemble d'étapes d'une représentation de la situation à une autre représentation.

Règle 4: Plusieurs transitions simultanément franchissables sont simultanément franchies.

4.5.6 Activation et désactivation simultanées d'une étape

Si une même étape participe à la description de la situation précédente et à celle de la situation suivante, elle ne peut, en conséquence, que rester active.

Règle 5: Si, au cours du fonctionnement, une étape active est simultanément activée et désactivée, alors elle reste active.

4.6 Événements d'entrée

4.6.1 Généralités

Les règles d'évolution montrent que seul un changement des valeurs des variables d'entrée est susceptible de provoquer l'évolution d'un diagramme grafcet. Ce changement, appelé

"événement d'entrée" doit être défini par la valeur antérieure et la valeur postérieure de toutes les variables d'entrées pour caractériser cet événement unique. Dans la pratique, on ne spécifie que des ensembles d'événements d'entrée caractérisés par le changement d'état (front montant ou front descendant) d'une ou plusieurs variables booléennes d'entrée.

NOTE Le front montant d'une variable logique, qui se note par le signe " $\uparrow$ " devant une variable booléenne, indique que ce front montant n'est vrai qu'au passage de la valeur 0 à la valeur 1 de la variable concernée. Le front descendant d'une variable logique, qui se note par le signe " $\downarrow$ " devant une variable booléenne, indique que ce front descendant n'est vrai qu'au passage de la valeur 1 à la valeur 0 de la variable concernée.

On dit que "l'événement se produit" à la date du changement d'état des variables d'entrée qui le caractérisent.

4.6.2 Spécification des événements d'entrée

La spécification des événements d'entrée se fait par une expression logique d'une ou plusieurs variables caractéristiques, souvent dans une condition de transition et plus rarement en vue de spécifier directement un événement interne (voir 4.7).

$\uparrow a$ EXEMPLE 1

L'expression " $\uparrow a$ " décrit l'ensemble de tous les événements d'entrée pour lesquels la valeur antérieure de la variable d'entrée "a" est 0 et sa valeur postérieure est 1, quelles que soient les valeurs des autres variables d'entrée du système.

$a \cdot \uparrow b$ EXEMPLE 2

L'expression " $a \cdot \uparrow b$ " décrit l'ensemble de tous les événements d'entrée pour lesquels la valeur postérieure de la variable d'entrée "a" est 1 et la valeur antérieure de la variable d'entrée "b" est 0 et sa valeur postérieure est 1, quelles que soient les valeurs des autres variables d'entrée du système.

a EXEMPLE 3

L'expression "a" décrit l'ensemble de tous les événements d'entrée pour lesquels la valeur postérieure de la variable d'entrée "a" est 1, quelles que soient les valeurs des autres variables d'entrée du système.

NOTE Utilisée dans une condition de transition, cette expression peut conduire à une évolution fugace (voir 3.12).

4.7 Événements internes

4.7.1 Généralités

Seuls certains événements d'entrée peuvent se produire à partir d'une situation donnée. La conjonction d'une situation et d'un événement d'entrée pouvant se produire à partir de celle-ci s'appelle un événement interne (voir 3.6). Cette notion est principalement utilisée par le concepteur pour conditionner une affectation de sortie à un ensemble d'événements internes (voir 4.8.3). La description d'un ensemble d'événements internes se fait par l'un des moyens suivants.

4.7.2 Événements internes décrits par l'activation d'une étape

L'activation d'une étape, notée de manière graphique (symbole 27), décrit l'ensemble des événements internes qui ont chacun pour conséquence l'activation de cette étape.

4.7.3 Événements internes décrits par la désactivation d'une étape

La désactivation d'une étape, notée de manière graphique (symbole 28), décrit l'ensemble des événements internes qui ont chacun pour conséquence la désactivation de cette étape.

4.7.4 Événements internes décrits par le franchissement d'une transition

Le franchissement d'une transition, notée de manière graphique (symbole 29), décrit l'ensemble des événements internes qui ont chacun pour conséquence le franchissement de cette transition.

4.8 Modes de sortie

4.8.1 Généralités

Les actions permettent d'établir le lien entre l'évolution du grafset et les sorties. Deux modes de sortie, mode continu ou mode mémorisé, décrivent comment les sorties dépendent de l'évolution et des entrées du système.

4.8.2 Mode continu (assignation sur état)

En mode continu c'est l'association d'une action à une étape qui permet d'indiquer qu'une variable de sortie à la valeur vraie si l'étape est active et si la condition d'assignation est vérifiée. La condition d'assignation est une expression logique de variables d'entrée et/ou de variables internes (voir symbole 22). Si l'une des conditions n'est pas remplie et sous réserve qu'aucune autre action relative à la même sortie ne satisfasse les conditions, la variable de sortie considérée prend la valeur fausse.

On appelle assignation le fait d'imposer les valeurs (vraie ou fausse) à des variables de sortie.

L'ensemble des assignations locales (relatives aux étapes actives à un instant donné) définit l'assignation de toutes les variables de sortie pour cette situation.

Règle d'assignation: pour une situation donnée, les valeurs des sorties relatives aux actions continues sont assignées:

- à la valeur vraie, pour chacune des sorties relatives aux actions associées aux étapes actives et pour lesquelles les conditions d'assignation sont vérifiées,
- à la valeur fausse, pour les autres sorties (qui ne sont pas assignées à la valeur vraie).

4.8.3 Mode mémorisé (affectation sur événement)

En mode mémorisé c'est l'association d'une action à des événements internes qui permet d'indiquer qu'une variable de sortie ou interne prend et garde la valeur imposée si l'un de ces événements se produit.

Des représentations explicites sont nécessaires (activation d'étape, désactivation d'étape, franchissement d'une transition, etc.) pour décrire l'association des actions aux événements.

La valeur d'une variable de sortie ou interne relative à une action mémorisée reste inchangée tant qu'un nouvel événement spécifié ne la modifie pas.

On appelle affectation le fait de mémoriser, à un instant donné, l'attribution d'une valeur déterminée d'une variable de sortie ou interne.

Règle d'affectation: la valeur d'une variable de sortie ou interne, relative à une action mémorisée et associée à un événement, est affectée à la variable indiquée si l'événement interne spécifié se produit; à l'initialisation la valeur de cette variable est fausse (variable booléenne) ou nulle (variable numérique).

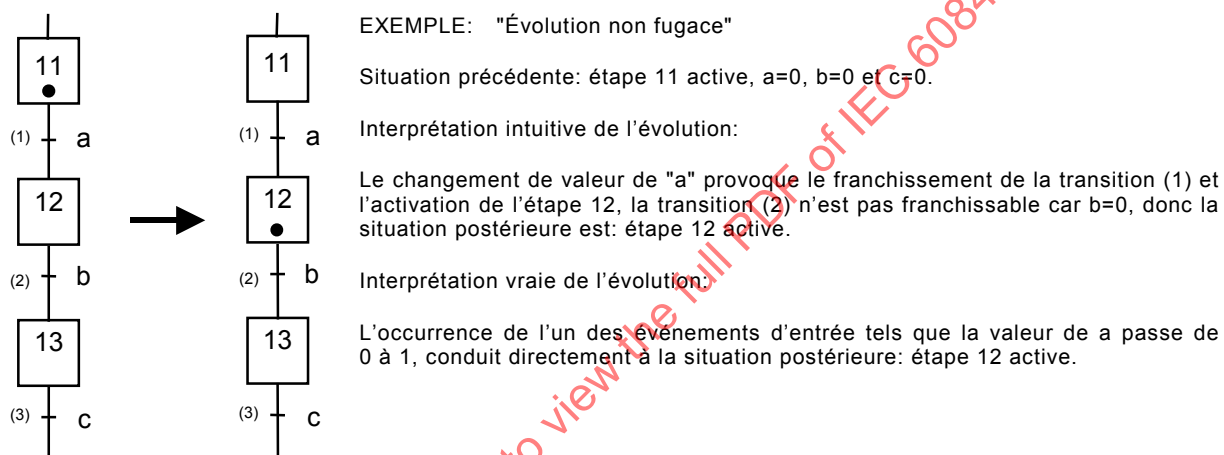
4.9 Application des règles d'évolution

4.9.1 Généralités

L'interprétation intuitive de l'évolution, dite "pas à pas", désigne la démarche progressive qui permet, sur occurrence d'un événement d'entrée et à partir de la situation précédente, de déterminer, par application successive des règles d'évolution sur chaque transition, la situation postérieure à l'événement considéré. Cette facilité d'interprétation est un artifice autorisant une spécification indirecte de l'évolution, mais le concepteur doit prendre garde au fait que le franchissement des transitions situées sur ce chemin n'implique pas l'activation effective des situations intermédiaires.

4.9.2 Évolution non fugace

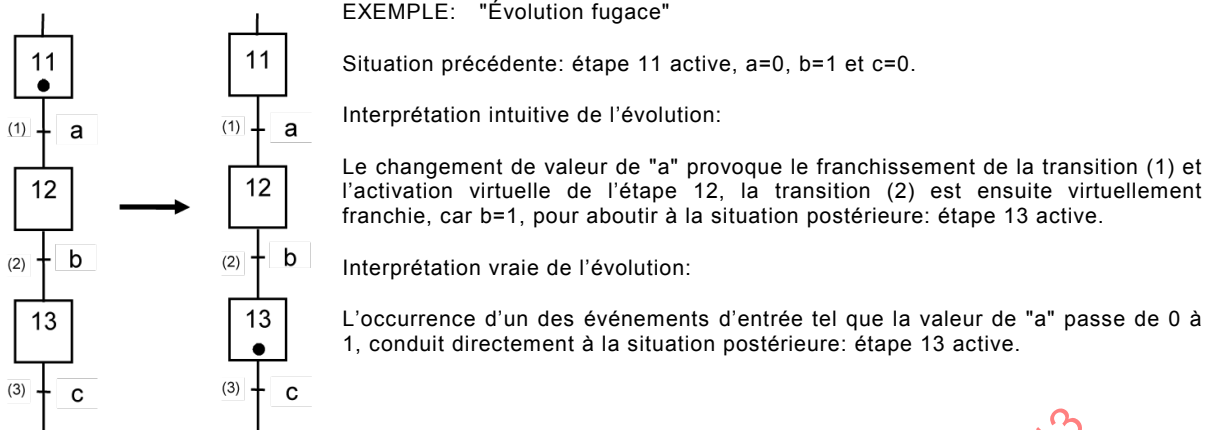
En règle générale, l'évolution est non fugace, c'est-à-dire que l'événement d'entrée ne provoque qu'un seul pas d'évolution (le franchissement simultané d'une ou plusieurs transitions).



4.9.3 Évolution fugace

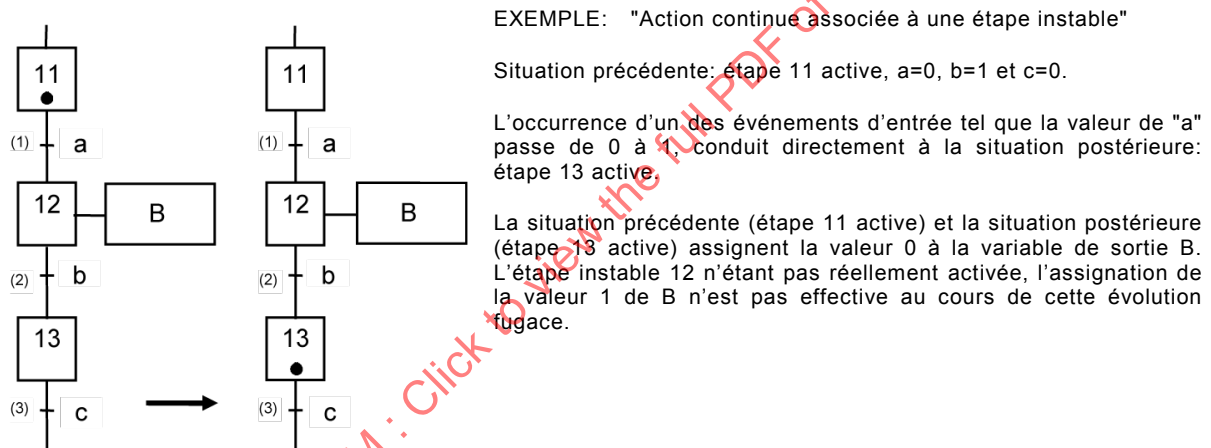
Dans certains cas, l'application des règles d'évolution peut conduire à franchir successivement des transitions (en plusieurs pas d'évolution) si les conditions de transition associées aux transitions postérieures sont déjà vraies lors du franchissement de la ou des premières transitions considérées. L'évolution correspondante, dite fugace, utilise le chemin parcouru pour indiquer comment on passe d'une situation précédente à une situation postérieure (voir 3.9).

Les étapes intermédiaires correspondantes, dites étapes instables, ne sont pas activées, mais on considère qu'elles ont été "virtuellement" activées et désactivées le long du chemin d'évolution intuitive, et de même les transitions correspondantes ont été "virtuellement" franchies.



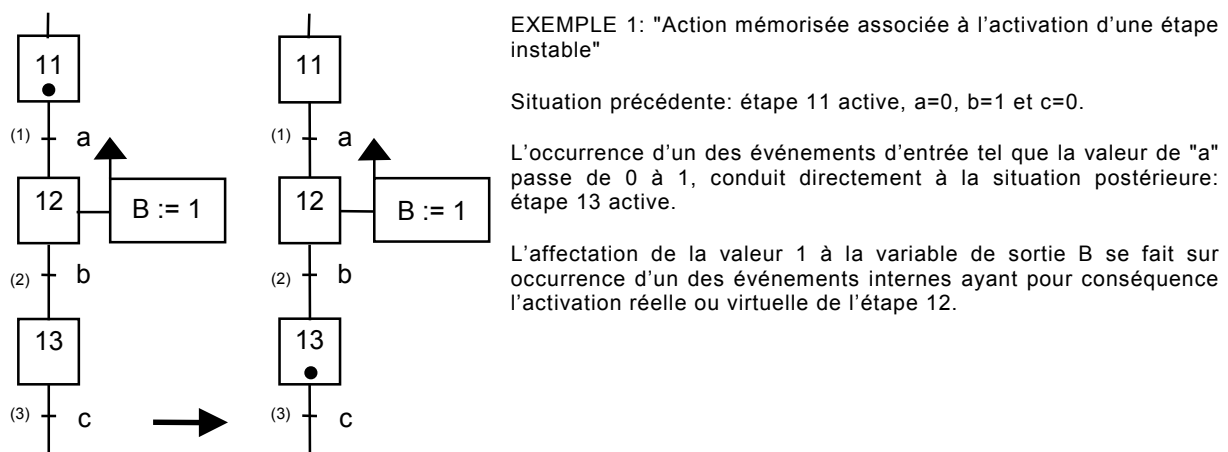
4.9.4 Conséquence d'une évolution fugace sur les assignations

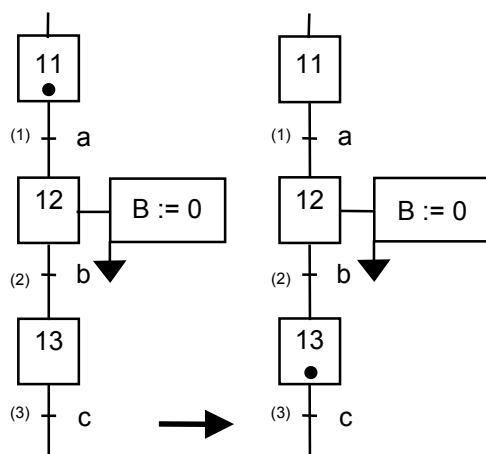
L'assignation d'une valeur de sortie par une action continue associée à une étape, qui à l'occasion d'une évolution fugace est une étape instable, n'est pas effective puisque l'étape n'est pas réellement activée (voir 4.8.2).



4.9.5 Conséquence d'une évolution fugace sur les affectations

L'affectation d'une valeur déterminée à une sortie par une action mémorisée (symbole 26) associée à une étape, qui à l'occasion d'une évolution fugace est une étape instable, est effective puisque cette affectation est associée aux événements qui déclenchent cette évolution (voir 4.8.3).





EXEMPLE 2: "Action mémorisée associée à la désactivation d'une étape instable"

Situation précédente: étape 11 active, $a=0$, $b=1$ et $c=0$.

L'occurrence d'un des événements d'entrée tel que la valeur de "a" passe de 0 à 1 conduit directement à la situation postérieure: étape 13 active.

L'affectation de la valeur 0 à la variable de sortie B se fait sur l'occurrence d'un des événements internes ayant pour conséquence la désactivation réelle ou virtuelle de l'étape 12.

4.10 Comparaison entre les deux modes de sortie

4.10.1 Généralités

Le choix du mode de sortie dépend des habitudes et des méthodologies, toutefois l'attention des concepteurs est attirée sur les importantes différences entre les deux modes.

4.10.2 Détermination de la valeur des sorties

Selon le mode choisi la détermination de la valeur des sorties est décrite de la manière suivante:

- en mode continu, toutes les sorties sont assignées selon la situation, à la valeur vraie pour les sorties désignées explicitement dans les actions associées aux étapes actives, à la valeur fausse pour les autres qui sont désignées implicitement par omission (voir règle d'assignation, 4.8.2);
- en mode mémorisé, seules les sorties affectées sont modifiées à la valeur indiquée, les valeurs des autres sorties mémorisées restent inchangées (voir règle d'affectation, 4.8.3).

4.10.3 Analyse de la valeur des sorties d'un diagramme grafcet à un instant déterminé

Selon le mode choisi l'analyse de la valeur des sorties est décrite de la manière suivante:

- en mode continu, la connaissance de la situation et de la valeur des entrées suffit pour déterminer la valeur des sorties (voir 4.8.2);
- en mode mémorisé, la connaissance de la situation et de la valeur des entrées ne suffit pas, il faut connaître également les évolutions précédentes pour déterminer la valeur des sorties (voir 4.8.3).

4.10.4 Actions relatives à une évolution fugace

Selon le mode choisi les actions relatives à une évolution fugace sont décrites de la manière suivante:

- en mode continu, les actions associées à une étape instable ne sont pas prises en compte car cette étape n'est pas activée (voir 4.9.2);
- en mode mémorisé, les actions associées à des événements correspondant à une évolution fugace sont prises en compte car les événements déclenchant cette évolution se produisent (voir 4.9.3).

4.10.5 Conflit éventuel sur la valeur des sorties

Selon le mode choisi le conflit éventuel sur la valeur des sorties est géré de la manière suivante:

- en mode continu, les principes de l'assignation permettent d'éviter tout conflit d'assignation sur une même sortie;
- en mode mémorisé, les principes de l'affectation ne permettent pas d'éviter les éventuels conflits d'affectation sur une même sortie. Le concepteur doit alors lui-même s'assurer que deux affectations contradictoires ne peuvent pas se produire simultanément.

NOTE 1 Les deux modes de sortie peuvent être utilisés dans une même spécification en GRAFCET, mais la valeur d'une variable de sortie est déterminée soit par assignation soit par affectation. La spécification d'une affectation sur une variable de sortie (mode mémorisé) exclut cette variable de toute assignation (mode continu).

NOTE 2 L'article 5 donne les symboles graphiques qui permettent de distinguer les actions continues (représentation par défaut) des actions mémorisées (représentation explicite selon l'ensemble des événements spécifiés).

NOTE 3 Dans le cas fréquent d'une spécification du comportement de la partie commande, l'usage industriel courant impose d'employer le mode continu pour toutes les sorties Booléenne à destination des actionneurs, et le mode mémorisé pour décrire des tâches internes de commande. Ces tâches, telles qu'incrémenter d'un compteur, ou modification de la valeur d'un registre numérique, portent sur des variables internes non nécessairement booléennes. Associées aux actions mémorisées, les tâches internes, ainsi que le calcul de prédicat associé aux conditions de transition, ne sont pas décrites dans la présente norme, mais sont associées par l'usage à la description logique des évolutions du grafcet.

5 Représentation graphique des éléments

Les éléments du GRAFCET possèdent une représentation symbolique qui permet, en les associant correctement, de réaliser des diagrammes fonctionnels clairs et synthétiques.

NOTE 1 Seule la représentation globale des symboles est imposée, les dimensions et les éléments de détail (épaisseur des traits, police de caractère, etc.) sont laissés libres aux utilisateurs.

NOTE 2 Les représentations en trait pointillé indiquent le contexte du symbole.

Tableau 1 – Étapes

N°	Symbole	Description
[1]		Étape: A un instant donné une étape est soit active, soit inactive. L'ensemble des étapes actives définit la situation du système à l'instant considéré. Le rapport longueur/largeur est arbitraire, bien qu'un carré soit recommandé. Les étapes sont identifiées par un repère, par exemple alphanumérique. L'astérisque au centre de la moitié supérieure du symbole général doit être remplacé par le repère attribué à l'étape. EXEMPLE 1: "Étape 2"  EXEMPLE 2: "Étape 3 représentée dans son état actif"  NOTE Il peut être utile d'indiquer quelles sont les étapes actives à un instant donné en marquant ces étapes par un point. Ce point n'appartient pas au symbole d'étape et est seulement employé à des fins d'explication.
[2.1]	X*	Variable d'étape: L'état actif ou inactif d'une étape peut être représenté respectivement par les valeurs logiques "1" ou "0" d'une variable booléenne X * dans laquelle l'astérisque * doit être remplacé par le repère de l'étape considérée. EXEMPLE: "Variable d'étape de l'étape 8" X8
[2.2]	T*	Durée d'étape: La durée d'une étape active peut être représentée par la valeur d'une variable temporelle T*, dans laquelle l'astérisque * doit être remplacé par le repère de l'étape considérée. EXEMPLE: "Durée d'étape de l'étape fill" TFill.
[3]		Étape initiale: Cette notation indique que cette étape participe à la situation initiale. NOTE 1 Les règles du symbole 1 sont applicables. NOTE 2 Une étape initiale peut-être "instable", voir 4.9.3. EXEMPLE: "Étape initiale 12" 
[4]		Étape encapsulante: Cette notation indique que cette étape contient d'autres étapes dites encapsulées. NOTE 1 Les règles du symbole 1 sont applicables. NOTE 2 Les propriétés et les exemples d'utilisation de l'étape encapsulante sont présentés en 7.4.
[5]		Étape encapsulante initiale: Cette notation indique que cette étape encapsulante participe à la situation initiale. NOTE Une étape encapsulante initiale contient au moins une étape encapsulée initiale.
[6]		Macro étape: Représentation unique d'une partie détaillée de GRAFCET, appelée expansion de la macro-étape. NOTE Les propriétés et les exemples d'utilisation de la macro-étape sont présentés en 7.5.

Tableau 2 – Transitions

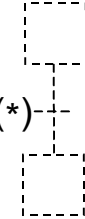
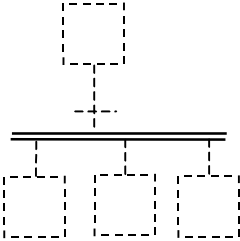
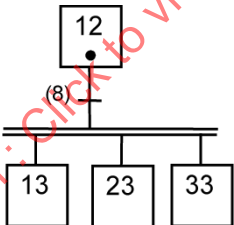
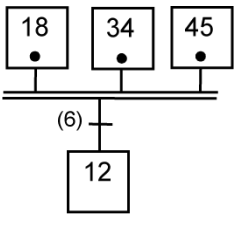
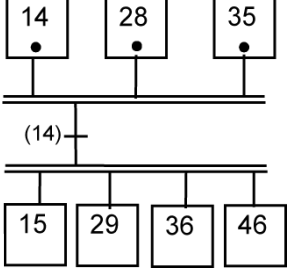
N°	Symbole	Description
[7]		Transition entre deux étapes: Une transition est représentée par un trait perpendiculaire aux liaisons joignant deux étapes. NOTE 1 La transition est validée lorsque l'étape amont est active (voir règle d'évolution No. 2, 4.5.3). NOTE 2 Il n'y a toujours qu'une seule transition entre deux étapes (voir 4.4). NOTE 3 Il est possible, pour des raisons de représentation graphique, de placer des transitions sur des segments de liaison horizontaux (voir Figure B.5, grafcet partiel G1). NOTE 4 Le symbolisme des transitions n'est pas traité dans la présente norme. Les transitions peuvent être décrites en clair, des expressions booléennes, des diagrammes logiques, etc.
[8]		Repère de transition: La transition peut comporter un repère, placé généralement à sa gauche, qu'il ne faut pas confondre avec la condition de transition associée. L'astérisque doit être remplacé par le repère alphanumérique de la transition.
[9]		Synchronisation en amont et/ou aval d'une transition: Lorsque plusieurs étapes sont reliées à la même transition, les liaisons orientées d'entrée et/ou de sortie de ces étapes sont regroupées en amont ou en aval par le symbole de synchronisation représenté par deux traits parallèles horizontaux. NOTE La référence du symbole de synchronisation est 9.2.2.5 de l'ISO 5807:1985.  EXEMPLE 1: Transition d'une étape (12) vers plusieurs (13, 23, 33). La transition (8) est validée lorsque l'étape (12) est active.  EXEMPLE 2: Transition de plusieurs étapes (18, 34, 45) vers une (12). La transition (6) n'est validée que lorsque toutes les étapes amont sont actives.  EXEMPLE 3: Transition de plusieurs étapes (14, 28, 35) vers plusieurs (15, 29, 36, 46). La transition (14) n'est validée que lorsque toutes les étapes amont sont actives.

Tableau 3 – Liaisons orientées

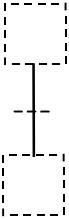
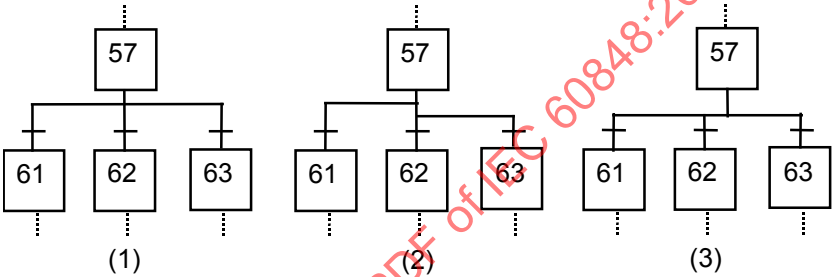
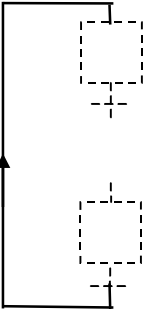
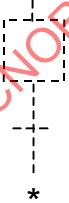
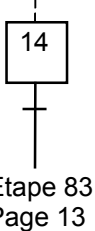
N°	Symbole	Description
[10]		Liaison orientée de haut en bas: Les voies d'évolution entre les étapes sont indiquées par des liaisons orientées reliant les étapes aux transitions et les transitions aux étapes. Les liaisons orientées sont horizontales ou verticales. Des liaisons obliques sont toutefois permises dans les cas exceptionnels où elles apportent plus de clarté au diagramme. Les croisements de liaisons verticales et horizontales sont admis s'il n'existe aucune relation entre ces liaisons. En conséquence, de tels croisements doivent être évités lorsque les liaisons correspondent à la même évolution. EXEMPLE: Les trois représentations sont admissibles, mais les représentations 2 et 3 sont recommandées pour éviter la confusion entre croisement sans et avec liaison. 
[11]		Liaison orientée de bas en haut: Par convention le sens d'évolution est toujours du haut vers le bas. Des flèches doivent être utilisées si cette convention n'est pas respectée ou si leur présence peut apporter une meilleure compréhension.
[12]		Repère de liaison: Lorsqu'une liaison orientée doit être interrompue (par exemple dans des dessins complexes ou dans le cas de représentation sur plusieurs pages), le repère des étapes de destination ainsi que le repère de la page à laquelle elle apparaît doivent être indiqués. L'astérisque doit être remplacé par le repère de liaison. EXEMPLE: Évolution vers l'étape 83 de la page 13. 

Tableau 4 – Conditions de transition associées

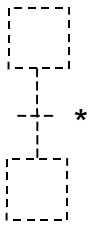
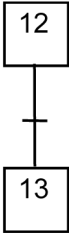
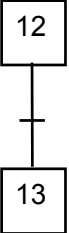
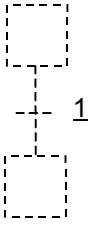
N°	Symbole	Description
[13]		Condition de transition: Une proposition logique, appelée condition de transition, qui peut être vraie ou fausse, est associée à chaque transition. S'il existe une variable logique correspondante, elle est égale à 1 quand la condition de transition est vraie et égale à 0 quand la condition de transition est fausse. La proposition logique formant la condition de transition est constituée d'une ou plusieurs variables booléennes (variable d'entrée, variable d'étape, valeur d'un prédicat, etc.). L'astérisque doit être remplacé par la description de la condition de transition associée à la transition sous forme d'un texte, d'une expression booléenne, ou à l'aide de symboles graphiques.  EXEMPLE 1: Description de la condition de transition par un texte Portillon fermé (a) et (pas de pression ($\overline{b}$) ou présence pièce (c))  EXEMPLE 2: Condition de transition décrite par une expression booléenne. $a . (\overline{b} + c)$
[14]		Condition de transition toujours vraie: La notation "1" indique que la condition de transition est toujours vraie. NOTE Dans ce cas, l'évolution est dite toujours fugace (voir 4.9.3), le franchissement de la transition n'est conditionné que par l'activité de l'étape amont.

Tableau 4 (suite)

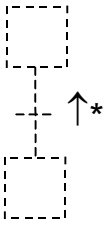
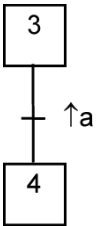
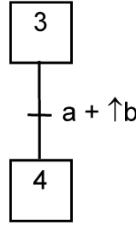
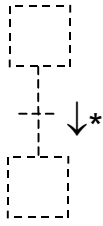
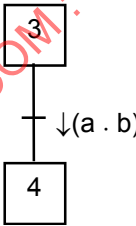
N°	Symbole	Description
[15]		Front montant d'une variable logique: La notation "$\uparrow$" indique que la condition de transition n'est vraie qu'au changement d'état de la variable * (front montant: passage de la valeur 0 à la valeur 1, voir la note de 4.6). Cette notation est générale et s'applique à toute proposition logique, qu'il s'agisse d'une variable élémentaire ou d'une combinaison de plusieurs variables booléennes.  EXEMPLE 1: La condition de transition associée n'est vraie que lorsque a passe de l'état 0 à l'état 1. NOTE En application de la règle d'évolution N°2, la transition n'est franchie que sur un front montant sur a postérieur à la validation de la transition par l'activité de l'étape 3.  EXEMPLE 2: La condition de transition associée n'est vraie que lorsque a est vraie ou que b passe de l'état 0 à l'état 1.
[16]		Front descendant d'une variable logique: La notation "$\downarrow$" indique que la condition de transition n'est vraie qu'au changement d'état de la variable * (front descendant: passage de la valeur 1 à la valeur 0, voir la note de 4.6). Cette notation est générale et s'applique à toute proposition logique, qu'il s'agisse d'une variable élémentaire ou d'une combinaison de plusieurs variables booléennes.  EXEMPLE: La condition de transition associée n'est vraie que lorsque le produit logique "a . b" passe de l'état 1 à l'état 0.

Tableau 4 (suite)

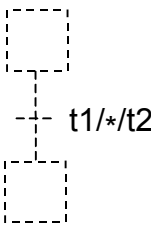
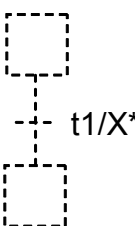
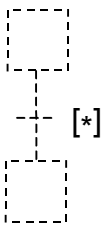
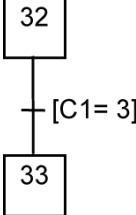
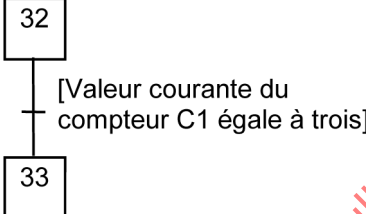
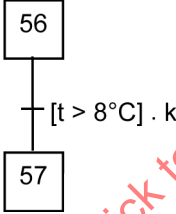
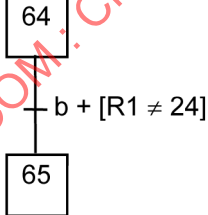
N°	Symbole	Description
[17]		Condition de transition dépendante du temps: La notation "t1/*/t2" indique que la condition de transition n'est vraie qu'après un temps t1 suivant l'occurrence du front montant ($\uparrow^*$) de la variable temporisée et redevient fausse après un temps t2 suivant l'occurrence du front descendant ($\downarrow^*$). L'astérisque doit être remplacé par la variable que l'on désire temporiser, par exemple une variable étape ou une variable entrée. t1 et t2 doivent être remplacés par leur valeur réelle exprimée dans l'unité de temps choisie. La variable temporisée doit rester vraie pendant un temps égal ou supérieur à t1 pour que la condition de transition puisse être vraie. NOTE Cette notation est celle de l'opérateur à retard défini par la CEI 60617-S01655 (2004-09). 14 $\downarrow$ 3s/a/7s 15 EXEMPLE: La condition de transition associée à la transition n'est vraie que 3 s après que "a" passe de l'état 0 à l'état 1, elle ne redevient fausse que 7 s après que "a" passe de l'état 1 à l'état 0.
[18]		Simplification usuelle du symbole 17: L'utilisation la plus courante est la temporisation de variable d'étape avec un temps t2 égal à zéro, ainsi la condition de transition devient fausse dès la désactivation de l'étape temporisée *. L'astérisque doit être remplacé par le repère de l'étape que l'on désire temporiser. L'étape temporisée doit rester active pendant un temps supérieur ou égal à t1 pour que la condition de transition puisse être vraie. Il est possible d'utiliser cette notation lorsque l'étape temporisée n'est pas l'étape amont de la transition. 27 $\downarrow$ 4s/X27 28 EXEMPLE: La condition de transition associée à la transition sera vraie 4 s après l'activation de l'étape 27, et sera fausse du fait du franchissement de la transition qui désactive l'étape amont. Dans ce cas, la durée d'activité de l'étape 27 est de 4 s.

Tableau 4 (suite)

N°	Symbole	Description
[19]		Valeur booléenne d'un prédicat: La notation "[*]" signifie que la valeur booléenne du prédicat constitue la variable de condition de transition. Ainsi, lorsque l'assertion * est vérifiée, le prédicat vaut 1, dans le cas contraire, il vaut 0. L'astérisque doit être remplacé par l'assertion que l'on veut tester. La variable booléenne de prédicat peut être associée à d'autres variables logiques pour constituer une proposition logique de condition de transition. EXEMPLE 1: La condition de transition associée est vraie lorsque l'assertion "C1 = 3" est vérifiée. 32  33 EXEMPLE 1a: La condition de transition associée est vraie lorsque la valeur courante du compteur C1 est égale à 3. NOTE: La forme de l'assertion n'est pas imposée, par exemple un langage littéral peut être utilisé. 32  33 EXEMPLE 2: La condition de transition associée est vraie lorsque l'assertion "t > 8 °C" est vérifiée et que la variable booléenne k vaut 1, c'est-à-dire lorsque la température t est supérieure à la valeur 8 °C et que le niveau haut k est atteint. 56  57 EXEMPLE 3: La condition de transition associée est vraie lorsque la variable booléenne b vaut 1 ou que l'assertion "R1 ≠ 24" est vérifiée, c'est-à-dire lorsque la pièce est à la place b, ou que le registre R1 n'a pas encore atteint la valeur 24. 64  65

Les symboles relatifs aux actions sont représentés avec des exemples dans le Tableau 5 et 6 ci-dessous. Les actions peuvent être des actions continues (Tableau 5) ou des actions mémorisées (Tableau 6).

Une action mémorisée possède un libellé (symbole 26) situé dans le rectangle qui décrit comment la variable de sortie est affectée à une valeur déterminée selon la règle d'affectation (voir 4.8.3).

La spécification des événements associés aux actions mémorisées est nécessaire pour indiquer quand l'affectation des sorties correspondantes se produit (voir règle d'affectation 4.8.3). Quatre moyens de description (symboles 27 à 29) permettent facilement de spécifier différents ensembles d'événements internes associés à des actions mémorisées.

Tableau 5 – Actions continues

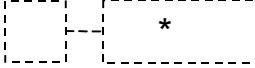
N°	Symbole	Description
[20]		Action continue: Une action continue est nécessairement associée à une étape. Plusieurs actions peuvent être associées à une même étape. Le rapport longueur/largeur est arbitraire, bien qu'un rectangle de hauteur égale à l'étape soit recommandé. A défaut d'une symbolisation explicite d'action mémorisée (symboles 27 à 29), le symbole général rectangulaire associé à une étape désigne toujours une action continue.
[21]		Libellé d'assignation d'une sortie: Toute action doit posséder un libellé dans le rectangle représentant cette action. Le libellé d'une action continue est la désignation de la variable de sortie assignée à la valeur vraie selon la règle d'assignation (voir 4.8.2). L'astérisque doit être remplacé par le libellé désignant la variable de sortie. L'expression littérale du libellé peut prendre une forme impérative ou une forme déclarative, seule compte la référence à la sortie. L'ordre dans lequel les actions sont représentées n'implique aucune séquence entre les actions. EXEMPLE 1: Différentes formes, littérales et symboliques, de libellé d'action faisant référence à la sortie dont la valeur vraie doit provoquer l'ouverture de la vanne 2. 4 Ouvrir vanne 2 4 vanne 2 4 YV2 EXEMPLE 2: Différentes représentations (1, 2, 3, 4) de l'association de plusieurs actions à une même étape. (1) 6 A B C (2) 6 A B C (3) 6 A B C (4) 6 A B C NOTE Les quatre représentations sont strictement équivalentes. Les représentations (2) et (4) peuvent être considérées respectivement comme des simplifications des représentations (1) et (3).

Tableau 5 (suite)

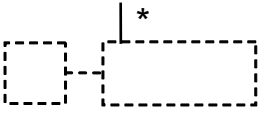
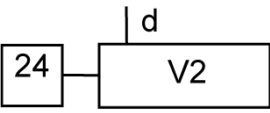
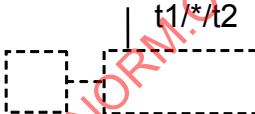
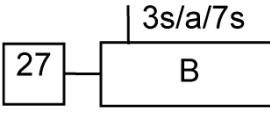
N°	Symbole	Description
[22]		Condition d'assignation: Une proposition logique, appelée condition d'assignation, qui peut être vraie ou fausse, conditionne toute action continue. L'absence de notation signifie que la condition d'assignation est toujours vraie. L'astérisque doit être remplacé par la description de la condition d'assignation sous forme d'un texte, ou d'une expression booléenne entre des variables d'entrées et/ou des variables internes. La condition d'assignation ne doit jamais comporter de front de variable (voir les symboles 15 et 16), car l'action continue n'est évidemment pas mémorisée, l'assignation sur événement n'ayant aucun sens (voir 4.8.3). EXEMPLE 1: La sortie V2 est assignée à la valeur vraie lorsque l'étape 24 est active et lorsque la condition d'assignation d est vraie. Dans le cas contraire, la sortie V2 est assignée à la valeur fausse. Autrement dit (sous forme d'une équation booléenne): $V2 = X24 \cdot d$  NOTE X24 est la variable d'étape reflétant l'activité de l'étape 24. EXEMPLE 2: La sortie V2 est assignée à la valeur vraie lorsque l'étape 24 est active (la condition d'assignation étant toujours vraie). Dans le cas contraire, la sortie V2 est assignée à la valeur fausse. Autrement dit (sous forme d'une équation booléenne): $V2 = X24$ 
[23]		Condition d'assignation dépendante du temps: La notation "t1/*t2" indique que la condition d'assignation n'est vraie qu'après un temps t1 suivant l'occurrence du front montant ($\uparrow^*$ voir symbole 15) de la variable temporisée * et redevient fausse après un temps t2 suivant l'occurrence du front descendant ($\downarrow^*$ voir symbole 16). L'astérisque doit être remplacé par la variable que l'on désire temporiser, par exemple une variable d'étape ou une variable d'entrée. t1 et t2 doivent être remplacés par leur valeur réelle exprimée dans l'unité de temps choisie. La variable temporisée doit rester vraie pendant un temps égal ou supérieur à t1 pour que la condition d'assignation puisse être vraie. NOTE Cette notation est celle de l'opérateur retard défini par la CEI 60617-S01655 (2004-09). EXEMPLE: La condition d'assignation n'est vraie que 3 s après que "a" passe de l'état "0" à l'état "1", elle redevient fausse 7 s après que "a" passe de l'état "1" à l'état "0".  La valeur de la sortie B dépend de l'activité de l'étape 27 et de la valeur de la condition d'assignation (voir règles d'assignation en 4.8.2).

Tableau 5 (suite)

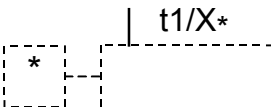
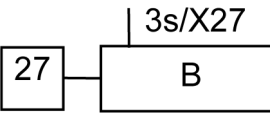
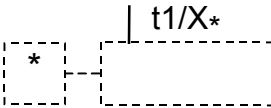
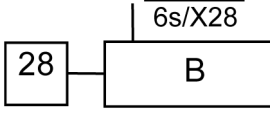
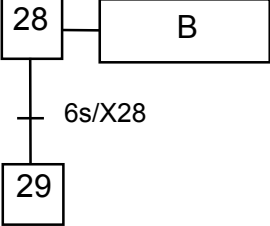
N°	Symbole	Description
[24]		Action retardée: L'action retardée est une action continue dont la condition d'assignation n'est vraie qu'après une durée t1 spécifiée suivant l'activation de l'étape associée *, dans le but de retarder l'assignation de la valeur vraie à la sortie correspondante. EXEMPLE: La sortie B est assignée à la valeur vraie lorsque 3 s se sont écoulées depuis l'activation de l'étape 27.  NOTE Si la durée d'activité de l'étape 27 est inférieure à 3 s, la variable de sortie B n'est pas assignée à la valeur vraie.
[25]		Action limitée dans le temps: L'action limitée dans le temps est une action continue dont la condition d'assignation n'est vraie que pendant une durée t1 spécifiée après l'activation de l'étape * à laquelle elle est associée, dans le but de limiter la durée d'assignation à la valeur vraie de la sortie correspondante. EXEMPLE 1: La sortie B n'est assignée à la valeur vraie que pendant 6 s suivant l'activation de l'étape 28.  NOTE Si la durée d'activité de l'étape 28 est inférieure à 6 s, la variable de sortie B est assignée à la valeur vraie uniquement pendant la durée d'activité de l'étape 28. Représentation équivalente: On peut, pour obtenir une limitation de la durée d'assignation à la valeur vraie de la sortie correspondante, utiliser l'opérateur retard simplifié dans la condition de transition associée de l'étape aval (voir symbole 18). EXEMPLE 2: Représentation équivalente de l'exemple 1 à l'aide du symbole 18. La sortie B n'est assignée à la valeur vraie que pendant 6 s suivant l'activation de l'étape 28. 

Tableau 6 – Actions mémorisées

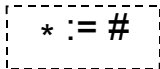
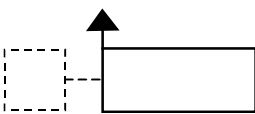
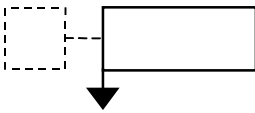
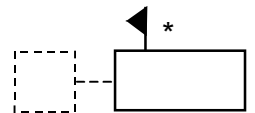
N°	Symbole	Description
[26]		Affectation de la valeur # à une variable *: Le libellé indique, dans une action mémorisée, la mise à la valeur # d'une variable *, lorsqu'un des événements associés à l'action se produit (voir règle d'affectation en 4.8.3). L'action mémorisée supportant cette affectation doit être associée à la description d'événements internes (symboles 27 à 29). L'affectation peut être décrite de manière littérale à l'intérieur du rectangle d'action. A := 1 EXEMPLE 1: Mise à la valeur vraie d'une variable booléenne A. Le libellé "A:= 1" permet de décrire l'affectation de la valeur 1 à une variable booléenne A lorsque l'un des événements associés à l'action se produit. b := 0 EXEMPLE 2: Mise à la valeur faussée d'une variable booléenne b. Le libellé "b:= 0" permet de décrire l'affectation de la valeur 0 à une variable booléenne b lorsque l'un des événements associés à l'action se produit. C := C+1 EXEMPLE 3: Incrémentation d'un compteur Le libellé "C:= C+1" permet de décrire l'affectation de la valeur C+1 à une variable numérique C lorsque l'un des événements associés à l'action se produit.
[27]		Action à l'activation: Une action à l'activation est une action mémorisée associée à l'ensemble des événements internes qui ont chacun pour conséquence l'activation de l'étape liée à cette action. La représentation traditionnelle de l'action par un rectangle est complétée, au côté gauche, d'une flèche symbolisant l'activation de l'étape. 37 B := 0 EXEMPLE: La variable booléenne B est affectée à la valeur 0 lorsque l'un des événements, conduisant à l'activation de l'étape 37, se produit.

Tableau 6 (suite)

N°	Symbole	Description
[28]		Action à la désactivation: Une action à la désactivation est une action mémorisée associée à l'ensemble des événements internes qui ont chacun pour conséquence la désactivation de l'étape liée à cette action. La représentation traditionnelle de l'action par un rectangle est complétée, au côté gauche, d'une flèche symbolisant la désactivation de l'étape. 24 K := 1  EXEMPLE: La variable booléenne K est affectée à la valeur 1 lorsque l'un des événements, conduisant à la désactivation de l'étape 24, se produit.
[29]		Action sur événement: Une action sur événement est une action mémorisée associée à chacun des événements internes décrits par l'expression *, à condition que l'étape, à laquelle l'action est reliée, soit active. La représentation traditionnelle de l'action par un rectangle est complétée, sur le côté haut, d'un symbole indiquant que l'action est conditionnée à l'occurrence d'un des événements internes spécifiés par l'expression *. Il est impératif que l'expression logique *, qui doit décrire un ensemble d'événements internes, comporte un ou plusieurs fronts de variables d'entrée. 13 H := 0  56 Q := 1  36 Z := 0  EXEMPLE 1: La variable booléenne H est affectée à la valeur 0 lorsque l'un des événements, représentés par "↑a" se produit et que simultanément l'étape 13 est active. NOTE L'ensemble d'événements d'entrée représentés par l'expression "↑a" combiné avec l'activité de l'étape 13, représente effectivement un ensemble d'événements internes (voir définition 3.6). EXEMPLE 2: la variable booléenne Q est affectée à la valeur 1 lorsque l'un des événements, représentés par "↑(a + b)" se produit et que simultanément l'étape 56 est active. EXEMPLE 3: la variable booléenne Z est affectée à la valeur 0 lorsque l'un des événements, représentés par "↑b . X28" se produit et que simultanément les étapes 36 et 28 sont actives.

Le Tableau 7 présente des commentaires relatifs aux éléments GRAFCET.

Tableau 7 – Commentaires associés aux éléments d'un grafcet

N°	Symbole	Description
[30]	"*"	Commentaire: Un commentaire relatif aux éléments graphiques d'un grafcet doit être placé entre guillemets. L'astérisque doit être remplacé par le commentaire. 45 "étape d'attente" 28 — E.P. "estampage pièce" 43 33 "synchronisation" — g 44 EXEMPLE 1: commentaire "étape d'attente" relatif à l'étape 45. EXEMPLE 2: commentaire "estampage pièce" relatif à l'action associée à l'étape 28. EXEMPLE 3: commentaire "synchronisation" relatif à une transition.

6 Représentation graphique des structures de séquences

6.1 Généralités

Le concepteur peut, sous réserve de l'application stricte de la règle de syntaxe imposant l'alternance étape/transition, réaliser des grafcet utilisant différentes structures caractéristiques.

6.2 Structures de base

6.2.1 Séquence



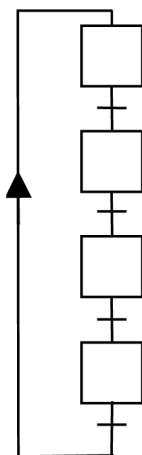
On appelle séquence une succession d'étapes telles que:

- chaque étape, excepté la dernière, ne possède qu'une seule transition aval,
- chaque étape, excepté la première, ne possède qu'une seule transition amont validée par une seule étape de la séquence.

NOTE 1 La séquence est dite "active" si au moins une de ses étapes est active, elle est dite "inactive" lorsque aucune de ses étapes n'est active.

NOTE 2 Le nombre d'étapes formant une séquence est aussi grand que l'on veut.

6.2.2 Cycle d'une seule séquence



Cas particulier de séquence rebouclée sur elle-même telle que:

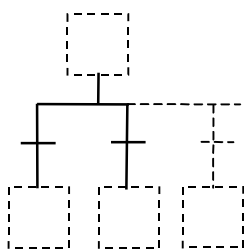
- chaque étape ne possède qu'une seule transition aval,
- chaque étape ne possède qu'une seule transition amont validée par une seule étape de la séquence.

NOTE 1 Un cycle d'une seule séquence peut constituer un grafcet partiel (voir 7.2.2).

NOTE 2 Pour permettre l'activation de ses étapes, un cycle d'une seule séquence doit satisfaire au moins à l'une des conditions suivantes:

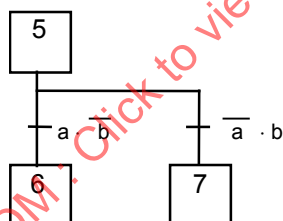
- posséder au moins une étape initiale,
- faire l'objet d'un ordre de forçage depuis un grafcet partiel de niveau supérieur (voir 7.3),
- appartenir à une des encapsulations d'une étape encapsulante (voir 7.4).

6.2.3 Sélection de séquences

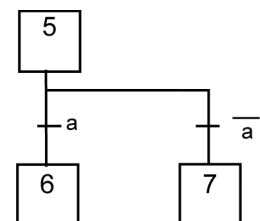


La sélection de séquences exprime un choix d'évolution entre plusieurs séquences à partir d'une ou de plusieurs étapes. Cette structure se représente par autant de transitions validées simultanément qu'il peut y avoir d'évolutions possibles.

NOTE L'exclusion entre les séquences n'est pas structurelle. Pour l'obtenir, le concepteur doit s'assurer soit de l'incompatibilité mécanique ou temporelle des conditions de transition, soit de leur exclusion logique.

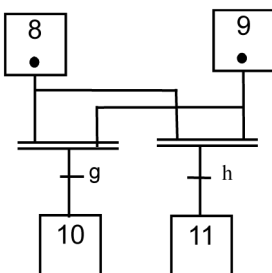


EXEMPLE 1: L'exclusion entre les séquences est obtenue par l'exclusion logique des deux conditions de transition, si "a" et "b" sont simultanément vraies lorsque l'étape 5 est active, aucune des transitions n'est franchissable.



EXEMPLE 2: Séquence prioritaire.

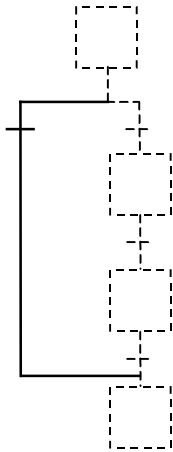
Dans cet exemple, une priorité est donnée à la transition 5/6, qui est franchie lorsque "a" est vraie.



EXEMPLE 3: Sélection de séquences à partir d'une synchronisation de deux séquences amont.

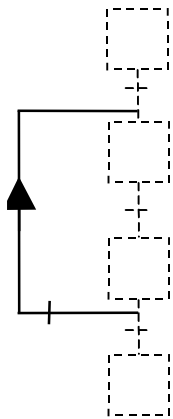
La sélection des séquences aval, par g et h, n'est possible que lorsque les deux transitions sont validées par l'activité simultanée des étapes 8 et 9 (symbole 9).

6.2.4 Saut d'étapes



Cas particulier de sélection de séquences, qui permet soit de parcourir la séquence complète soit de sauter une ou plusieurs étapes de la séquence lorsque, par exemple, les actions associées à ces étapes deviennent inutiles.

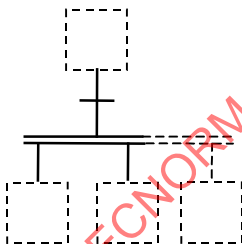
6.2.5 Reprise de séquence



Cas particulier de sélection de séquences, qui permet de recommencer la même séquence jusqu'à ce que, par exemple, une condition fixée soit obtenue.

NOTE Il est possible, pour des raisons de représentation graphique, de placer des transitions sur des segments de liaison horizontaux (voir Note 3 du symbole 7).

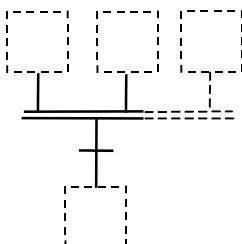
6.2.6 Activation de séquences parallèles



Le symbole 9 de synchronisation est utilisé dans cette structure pour indiquer l'activation simultanée de plusieurs séquences à partir d'une ou plusieurs étapes.

NOTE Après leur activation simultanée, l'évolution des étapes actives dans chacune des séquences parallèles devient alors indépendante.

6.2.7 Synchronisation de séquences



Le symbole 9 de synchronisation est utilisé dans cette structure pour indiquer l'attente de la fin des séquences amont avant d'activer la séquence aval.

NOTE La transition n'est validée que lorsque toutes les étapes amont sont actives.