**PROJET Fi3 :** ***aide à la conception de plans de maison et visite virtuelle.***

**SUJET**

Le projet consistera à développer un outil d'aide à la conception de plan de maison comportant les aspects suivants:
(1) À partir d'une demande sur la maison à construire (qui sera donnée sous la forme de contraintes) le système devra produire un ou plusieurs plans vérifiant la demande initiale qui pourront être visualisés en 2D (par exemple en produisant du code TikZ). Cette partie sera faite en utilisant Prolog (et ses contraintes).
(2) Il devra ensuite fournir un parcours de visite des différentes pièces de la maison. Le parcours de visite sera fait en utilisant Prolog. Le parcours de visite essayera dans la mesure du possible que chaque pièce ne soit visitée qu'au plus qu'une fois, et que toutes les pièces d'un même étage soient visitées de manière consécutive (c'est-à-dire sans changer d'étage). Une demande comportera le type de contraintes suivantes:
(1) Le fait que tout se fait sur un étage (minimum demandé) ou sur deux étages.
(2) La surface au sol disponible (surface sur un étage) ainsi qu'un encadrement sur la largueur et longueur minimum de la maison.
(3) Pour chaque pièce de la maison qui peut prendre la forme d'un rectangle ou d'un L (*i.e., deux rectangles accolés dont deux bords correspondent*), l'éventuel étage où elle doit se situer, sa surface minimum et maximum, ses dimensions et son type (voir figure ci-dessous comment on donne les tailles d'une pièce), et son orientation éventuelle (nord, sud, est, ouest), le fait qu'elle doit éventuellement être située directement sur une façade ou pas.
(4) Des contraintes d'accessibilité entre deux pièces données (il faut qu'il y ai une porte dont la largueur soit comprise entre telle et telle valeur entre deux pièces données). L'ensemble de ces contraintes d'accessibilité doit garantir l'accès à toutes les pièces.

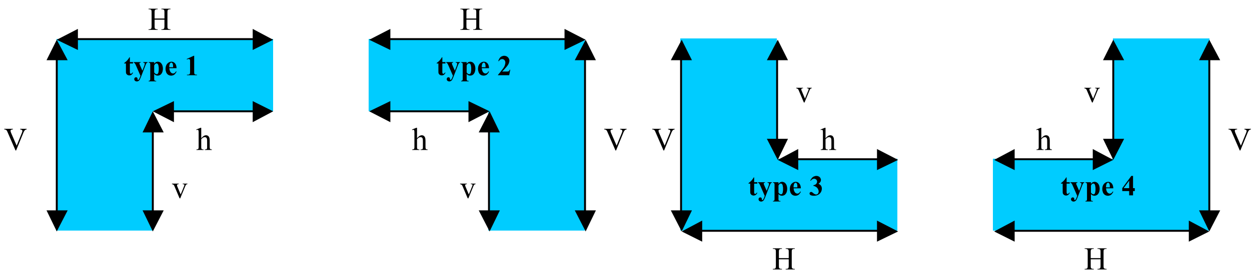


**Figure 1 : définition des tailles d'une pièce** (*h=v=0 dans le cas d'un rectangle*)

L'annexe 1 comporte une description de la problématique générale avec quelque références bibliographique tandis que l'annexe 2 fournir un exemple de format minimum à utiliser pour représenter les contraintes d'un problème. Si vous le souhaitez, vous pourrez étendre ce format en ne touchant pas aux prédicats existant, mais en ajoutant d'autres à votre convenance pour rajouter d'autres types de contraintes. **Il est impératif que vous traitiez l'exemple donné en annexe ainsi qu'un exemple vous étant propre avec au moins une pièce en forme de L.**

**Annexe 1 : Description générale de la problématique et exemple de problème type**

## 2.7.92 Floor planning problem

- `diffn`,
- `geost`,
- `lex_chain_less`.

A constraint that can be used for the *floor planning problem*. The *floor planning problem* [236, 205, 211] involves various type of spaces such as the *placement space* itself (i.e., the *floor*), the *rooms* to place within the placement space and the *circulation* between the rooms. The placement space can be located on one single level or on several levels. Very often the placement space corresponds to one single rectangle and all rooms are rectangles with their borders parallel to the contour of the placement space. Circulation typically corresponds to *corridors* or *stairs* that respectively allow to access from one room to another room or from one level to another level. Within the context of floor planning three main classes of constraints have been identified [212], namely dimensionals, topological and implicit constraints:

- A *dimensional constraint* usually restricts the length, the width or the surface of *one single space*. Ratio constraints enforce aesthetic proportions between the length and the width of a single space or constraint the surfaces of two closely related spaces such as, for instance, the toilets and the shower. Dimensional constraints can be expressed by reducing the domain of some variable or by stating some arithmetic constraints between two variables.

- A *topological constraint* imposes a condition between *two spaces*. Typical topological constraints are:
  - *Adjacency constraints with a minimum contact* between a room and a corridor or an other room allow expressing the fact that their must be enough place to put a door between two given spaces. In the context of staircases one has to enforce that fact that the first and last stairs are completely accessible. When a corridor is made up from two parts, one has also to enforce the fact that the two parts are fully in contact.
  - *Adjacency with the contour constraints* between a room and a specified (or not) side of the contour allow expressing the orientation of a room (or just the fact that a room must have some window).
  - *Relative positioning constraints* between two specified rooms allow for instance expressing the fact that a room is located to the north of another room.
  - *Minimum and maximum distance constraints* between two rooms allow expressing the proximity between two given rooms.

  Topological constraints occur naturally in the preliminary design phase in architecture and can typically be expressed by using reified or global constraints.

- An *implicit constraint* puts a global condition that is inherent to floor planning problems between *all the spaces* of the floor. We typically have:
  - *Inclusion* of each room and circulation within the contour.
  - *Partitioning* of the placement space (i.e., no wasted space is permitted). This is usually a hard constraint which requires specific propagation in order to prevent the creation of wasted space.
  - *Non-overlapping* between rooms.
  - *Symmetry breaking constraints* between identical rooms imposes for instance a lexicographic order between their respective lower leftmost corners.

  Such constraints can typically be expressed by using global constraints such as `diffn`, `geost` or `lex_chain_less`.

Finally, in order to allocate as much surface as possible to the rooms, one wants sometimes to minimise the total circulation area between the different rooms.

In order to illustrate these constraints we now consider an example of floor planning problem taken from R. Maculet PhD thesis [205] involving 11 spaces. Constraints on the dimensions of these space are:

- The *floor* where to place everything has a size of 10 by 12 meters.

- The *living* has a surface between 33 and 42 square meters and a minimum size of 4 by 4.

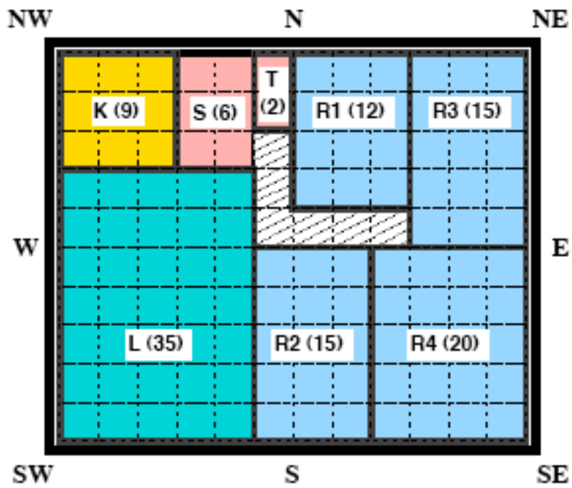


Figure 2.18: A solution to Maculet floor planning problem which minimises the total area of the corridors

- The *kitchen* has a surface between 9 and 15 square meters and a minimum size of 3 by 3.
- The *shower* has a surface between 6 and 9 square meters and a minimum size of 2 by 2.
- The *toilet* has a surface between 1 and 2 square meters and a minimum size of 1 by 1.
- The *first and second parts of the corridor* have both a surface between 1 and 12 square meters and a minimum size of 1 by 1.
- The *first, second and third rooms* have all a surface between 11 and 15 square meters and a minimum size of 3 by 3.
- The *fourth room* has a surface between 15 and 20 square meters and a minimum size of 3 by 3.

Topological constraints between spaces are:

- The *living* is located on the south-west contour. The *kitchen*, the *first*, *second* and *third rooms* are either located on the south or on the north contour. The *fourth room* is on the south contour.
- All spaces, except the *kitchen*, are adjacent to one of the *corridors* with at least 1 meter of full contact.
- The *kitchen* is adjacent to the *living* and to the *shower*.
- The *toilet* is adjacent to the *kitchen* or to the *shower*.
- The *first* and the *second parts* of the *corridor* are adjacent and fully in contact.

Finally no wasted space is permitted. Figure 2.18 presents a solution to the corresponding floor planning problem that minimises the area of the two corridors.

[205] R. Maculet. *Représentation des connaissances spatiales (algèbre de Manhattan et raisonnement spatial avec contraintes*. PhD thesis, Paris VI University, France, 1991. In French. 157, 158

[211] B. Medjdoub. *Méthodes de conception fonctionnelle en architecture: une approche CAO basée sur des contraintes: ARCHiPLAN*. PhD thesis, Ecole Centrale de Paris, France, 1996. In French. 157

[212] B. Medjdoub and B. Yannou. Separating topology and geometry in space planning. *Computer-aided design*, 32(1):39–61, 2000. 157

[236] C. E. Pfefferkorn. A Heuristic Problem Solving Design System for Equipment or Furniture Layouts. *Communications of the ACM*, 18(ISSN:0001-0782):286–297, May 1975. 157

## Annexe 2 : Format minimum pour représenter un problème

Cette annexe donne un exemple de représentation du problème mentionné dans l'annexe 1 sous forme d'un ensemble de prédicats.

```
% PROBLEM INSTANCE (from Robert Maculet PhD thesis)

% DECLARATION OF THE DIFFERENT SPACE WITH THEIR RESPECTIVE DIMENSIONAL CONSTRAINTS
% space(kind, name, min_H, max_H, min_V, max_V, min_h, max_h, min_v, max_v, min_ratio, max_ratio, min_surf, max_surf).
% (first a floor fact for which all sizes are fixed)
space(floor    , home     , 12, 12,  0, 0, 10, 10, 0, 0, _, _,  120,  120).
space(room     , living   ,  4,  _,  0, 0,  4,  _, 0, 0, _, _,   33,   42).
space(room     , room1    ,  3,  _,  0, 0,  3,  _, 0, 0, _, _,   11,   15).
space(room     , room2    ,  3,  _,  0, 0,  3,  _, 0, 0, _, _,   11,   15).
space(room     , room3    ,  3,  _,  0, 0,  3,  _, 0, 0, _, _,   11,   15).
space(room     , room4    ,  3,  _,  0, 0,  3,  _, 0, 0, _, _,   15,   20).
space(room     , kitchen  ,  3,  _,  0, 0,  3,  _, 0, 0, _, _,    9,   15).
space(room     , shower   ,  2,  _,  0, 0,  2,  _, 0, 0, _, _,    6,    9).
space(room     , toilet   ,  1,  _,  0, 0,  1,  _, 0, 0, _, _,    1,    _).
space(corridor, corridor1,  1,  _,  0, 0,  1,  _, 0, 0, _, _,    1,   12).
space(corridor, corridor2,  1,  _,  0, 0,  1,  _, 0, 0, _, _,    1,   12).

% DECLARATION OF THE ORIENTATIONS CONSTRAINTS
% contour(room or corridor, list of directions: n, s, o, e, no, so, ne, se).
contour(living  , [so]).
contour(kitchen, [s,n]).
contour(room1  , [s,n]).
contour(room2  , [s,n]).
contour(room3  , [s,n]).
contour(room4  , [s]).

% DECLARATION OF THE ADJACENCY CONSTRAINTS
% adj(room or corridor, list of rooms or corridors).
adj(living    ,   [corridor1,corridor2]).
adj(shower    ,   [corridor1,corridor2]).
adj(toilet    ,   [corridor1,corridor2]).
adj(room1     ,   [corridor1,corridor2]).
adj(room2     ,   [corridor1,corridor2]).
adj(room3     ,   [corridor1,corridor2]).
adj(room4     ,   [corridor1,corridor2]).
adj(kitchen   ,   [living]              ).
adj(kitchen   ,   [shower]              ).
adj(toilet    ,   [kitchen,shower]      ).
adj(corridor1,   [corridor2]           ).
```