Vers une acceptabilité graduelle des arguments dans les systèmes d'argumentation

C. Cayrol
M.C. Lagasquie-Schiex

Décembre 2002

Rapport IRIT 2002-42-R

Résumé

L'argumentation est basée sur l'échange et l'évaluation d'arguments interagissant, puis sur la définition d'arguments acceptables en fonction de l'évaluation proposée. Dans ce document, en partant du cadre de travail proposé par [Dun95], qui énumère un certain nombre d'acceptabilités collectives, et en utilisant les évaluations graduelles que nous avons proposées dans [CLS01, CLS02], nous nous sommes consacrées à la gradualité dans la notion d'acceptabilité d'arguments.

Table des matières

1 Introduction **1**

2 Acceptabilité collective de [Dun95] **3**
2.1 Le cadre de [Dun95] et sa représentation graphique . . . 3
2.2 Acceptabilité collective . . . 4
2.3 Différents niveaux d'acceptabilité d'un argument . . . 5
2.3.1 Niveaux de base . . . 5
2.3.2 Premier raffinement des niveaux de base . . . 6
2.3.3 Quelques cas particuliers . . . 6

3 Évaluation graduelle des interactions **9**
3.1 L'approche "locale" de [CLS01] (évaluation générique) . . . 9
3.2 L'approche "globale" de [CLS01] (évaluation par tuples) . . . 10
3.2.1 L'étiquetage d'un argument par un tuple . . . 11
3.2.2 Comparaison d'arguments à l'aide de tuples . . . 12
3.2.2.1 Les idées utilisées . . . 12
3.2.2.2 L'algorithme de comparaison . . . 12
3.2.3 Quelques propriétés . . . 13
3.3 Différences essentielles entre évaluations "locale" et "globale" . . . 15

4 Vers une acceptabilité graduelle **17**
4.1 Comparaison basée sur l'évaluation graduelle . . . 17
4.2 Situation d'un argument par rapport à ses contrariants . . . 17
4.3 Compatibilité entre acceptabilité collective et évaluation graduelle . . . 18
4.3.1 Exemples montrant la non compatibilité dans le cas général . . . 18
4.3.2 Cas particuliers menant à la compatibilité . . . 19
4.3.2.1 Premier cas . . . 19
4.3.2.2 Second cas . . . 20

5 Conclusion **25**

Bibliographie **27**

Chapitre 1

Introduction

[Dun95] a montré que le cadre de l'argumentation constitue un outil puissant permettant aussi bien l'étude de nombreux systèmes formels de raisonnement de sens commun que la définition d'une sémantique pour les programmes logiques. L'argumentation est basée sur l'échange et l'évaluation d'arguments supportant des opinions, des assertions. On trouve des applications notamment dans le domaine juridique, dans les systèmes d'aide à la prise de décision collective ou d'aide à la négociation. La caractéristique fondamentale d'un système d'argumentation est la présence d'interactions et notamment de relations de contrariété entre les arguments avancés. Si l'argument prend par exemple la forme d'une preuve logique, on peut avancer des arguments pour une proposition et des arguments contre cette proposition, *i.e.* pour la proposition contraire.
Le processus d'argumentation comporte donc une étape d'évaluation de la force relative des arguments en présence, l'objectif final étant de sélectionner les arguments les plus acceptables en fonction de l'évaluation choisie. On distingue :
- une évaluation dite *intrinsèque* qui évalue un argument indépendamment des interactions avec les autres arguments. On peut ainsi exprimer à quel point l'argument augmente la confiance en l'assertion qu'il supporte. Cette évaluation peut prendre différentes formes (voir [KAEF95, Par97, PS97, AC98]).
- une évaluation *des interactions* selon laquelle un argument est évalué en fonction de ses "contrariants", des contrariants de ses contrariants (ses "défenseurs"), ... [1]. Plusieurs approches ont été proposées (voir [Dun95, AC98, JV99, BH01, CLS01, CLS02][2]) qui se distinguent par la richesse de l'ensemble des valeurs disponibles pour évaluer un argument.

Évaluation intrinsèque et prise en compte des interactions ont très souvent été utilisées séparément, selon les applications envisagées. On trouve cependant quelques travaux qui proposent une combinaison de ces deux critères (voir par exemple [AC98] et dans une moindre mesure [CLS01]).
Actuellement, la majorité des travaux sur l'acceptabilité des arguments consiste à exploiter la prise en compte des interactions en considérant que l'acceptabilité d'un argument est définie par son appartenance à un certain type d'ensembles (les ensembles définis comme *acceptables*) ; on parle d'*acceptabilité collective*. Le cadre de [Dun95] est tout à fait approprié à une telle approche mais conduit à des états binaires : l'argument est accepté ou non.

Notre objectif est d'introduire une gradualité dans cette notion d'acceptabilité de manière à distinguer entre arguments plus ou moins acceptables et ainsi définir des "niveaux d'acceptabilité". Pour cela, nous allons chercher à exploiter les évaluations graduelles des interactions, par la prise en compte de la "qualité" des contrariants, des défenseurs, ...
Dans la section 2 page 3, nous nous placerons dans le cadre défini par [Dun95] : celui d'un système d'argumentation constitué d'un ensemble d'arguments et d'une relation binaire sur cet ensemble. On utilisera une représentation graphique des systèmes d'argumentation. Cela nous permettra de présenter des notions d'acceptabilité collective proposées par [Dun95] et d'identifier les différents niveaux d'acceptabilité collective que l'on peut ainsi atteindre, en particulier, quand on prend en compte la situation d'un argument par rapport à celle de ses contrariants.
Puis, dans la section 3 page 9, nous rappellerons deux types d'évaluation graduelle issus de [CLS01] : une

[1] Nous ne prendrons en compte ici que les interactions dues à la notion de contrariété entre arguments ! Il peut exister d'autres types d'interaction (par exemple des arguments qui se renforceraient au lieu de se contrarier).
[2] [CLS01] est la version longue de [CLS02].

approche locale qui généralise certains travaux existants ([JV99, BH01]) et une approche globale introduite dans [CLS01].
Nous proposerons alors, dans la section 4 page 17, une notion d'acceptabilité graduelle basée sur les évaluations graduelles : on déduit des évaluations graduelles une comparaison des arguments que l'on confronte aux niveaux d'acceptabilité collective.

Chapitre 2

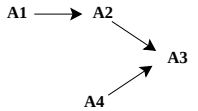
Acceptabilité collective de [Dun95]

2.1 Le cadre de [Dun95] et sa représentation graphique

Nous nous plaçons dans le cadre abstrait défini par [Dun95]. Soit le *système d'argumentation* $<\mathcal{A}, \mathcal{R}>$, $\mathcal{A}$ étant un ensemble d'arguments et $\mathcal{R}$ une relation binaire sur $\mathcal{A}$ appelée *relation de contrariété* : soit A_i et $A_j \in \mathcal{A}$, $A_i \mathcal{R} A_j$ signifiera que A_j est contrarié par A_i, ou que A_i contrarie A_j (aussi noté $(A_i, A_j) \in \mathcal{R}$). Un système d'argumentation sera dit *bien-fondé* si et seulement s'il n'existe pas de séquence infinie A_0, A_1, ..., A_n, ... telle que $\forall i, A_i \in \mathcal{A}$ et $A_{i+1} \mathcal{R} A_i$.
Nous ne préciserons pas davantage le format des arguments, ni la relation de contrariété.

Notations : Soit $A \in \mathcal{A}$, l'ensemble $\{A_i \in \mathcal{A} | A_i \mathcal{R} A\}$ est noté $\mathcal{R}^-(A)$ et l'ensemble $\{A_i \in \mathcal{A} | A \mathcal{R} A_i\}$ est noté $\mathcal{R}^+(A)$. $<\mathcal{A}, \mathcal{R}>$ définit un graphe orienté $\mathcal{G}$ (dit *graphe des contrariétés*).

Exemple : Le système $<\mathcal{A} = \{A_1, A_2, A_3, A_4\}, \mathcal{R} = \{(A_2, A_3), (A_4, A_3), (A_1, A_2)\}>$ définit le graphe $\mathcal{G}$ suivant ayant A_3 pour racine :



Définition 1 (Représentation graphique du système d'argumentation) *Soit $\mathcal{G}$ le graphe des contrariétés associé à un système d'argumentation $<\mathcal{A}, \mathcal{R}>$, on définit :*

Feuille du graphe des contrariétés *Un argument $A \in \mathcal{A}$ tel que $\mathcal{R}^-(A) = \varnothing$ sera une* feuille de $\mathcal{G}$.

Chemin dans le graphe des contrariétés *Un* chemin de A vers B *est une suite d'arguments $\mathcal{C} = A_1 - \ldots - A_n$ telle que :*
- $A = A_1$,
- $A_1 \mathcal{R} A_2$,
- ...,
- $A_{n-1} \mathcal{R} A_n$,
- $A_n = B$.

La longueur de ce chemin *est alors $n-1$ (le nombre d'arcs constituant ce chemin) et sera notée $l_{\mathcal{C}}$. L'ensemble des chemins de A vers B sera noté $\mathcal{C}(A, B)$.*

Dépendance, indépendance, racine-dépendance d'un chemin *Soit 2 chemins $\mathcal{C}_A \in \mathcal{C}(A_1, A_n)$ et $\mathcal{C}_B \in \mathcal{C}(B_1, B_m)$.*
Ces deux chemins seront dits dépendants *ssi $\exists A_i \in \mathcal{C}_A$, $\exists B_j \in \mathcal{C}_B$ tel que $A_i = B_j$.* Indépendants *sinon.*
Ces deux chemins seront dits racine-dépendants en A_n *ssi $A_n = B_m$ et $\forall A_i \neq A_n \in \mathcal{C}_A$, $\nexists B_j \in \mathcal{C}_B$ tel que $A_i = B_j$.*

Circuits dans le graphe des contrariétés *Un* circuit[1] *est un chemin $\mathcal{C} = A_1 - \ldots - A_n - A_1$ tel que $\forall i, j \in [1, n], i \neq j, \nexists A_i, A_j \in \mathcal{C}$ tels que $A_i = A_j$.*
Un circuit est isolé *quand aucun des arguments le composant n'a d'attaquant en dehors du circuit.*
Deux circuits $\mathcal{C}_A = A_1 - \ldots - A_n - A_1$ et $\mathcal{C}_B = B_1 - \ldots - B_m - B_1$ sont interconnectés *ssi $\exists i \in [1, n], \exists j \in [1, m]$ tels que $A_i = B_j$.*

[1] Cette définition d'un circuit correspond à la définition d'un circuit élémentaire en théorie des graphes (ne contient pas 2 arcs ayant la même origine, ou le même but).

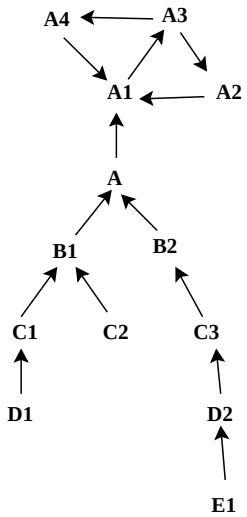
Nous allons utiliser les notions d'attaque et de défense directe ou indirecte inspirées par [Dun95] mais légèrement modifiées par nos soins[2]. On a donc :

Définition 2 (Attaquants/Défenseurs directs/indirects d'un argument) *Soit $A \in \mathcal{A}$:*
- *Les* attaquants directs *de A sont les éléments de $\mathcal{R}^-(A)$.*
- *Les* défenseurs directs *de A sont les attaquants directs des éléments de $\mathcal{R}^-(A)$.*
- *Les* attaquants indirects *de A sont les éléments A_i définis par :*
$$\exists \mathcal{C} \in \mathcal{C}(A_i, A) \text{ tel que } l_{\mathcal{C}} = 2k+1, \text{ avec } k \geq 1.$$
- *Les* défenseurs indirects *de A sont les éléments A_i définis par :*
$$\exists \mathcal{C} \in \mathcal{C}(A_i, A) \text{ tel que } l_{\mathcal{C}} = 2k, \text{ avec } k \geq 2.$$

On dira plus généralement que, si l'argument A est un attaquant (direct ou indirect) de l'argument B, alors A *attaque* B (ou B *est attaqué par* A). De même, si l'argument A est un défenseur (direct ou indirect) de l'argument B, alors A *défend* B (ou B *est défendu par* A).

Définition 3 (Branches d'attaque et de défense d'un argument) *Soit $A \in \mathcal{A}$, une* branche d'attaque *(resp.* de défense*) pour A est un chemin dans $\mathcal{G}$ d'une feuille vers A de longueur impaire (resp. paire). On dira alors que A est* racine d'une branche d'attaque *(resp.* de défense*).*

Toutes ces notions sont illustrées sur l'exemple suivant :



Sur ce graphe $\mathcal{G}$, on a donc (entre autres) :
- un chemin de C_2 vers A de longueur 2 ($C_2 - B_1 - A$),
- 2 circuits $A_1 - A_3 - A_2 - A_1$ et $A_1 - A_3 - A_4 - A_1$, chacun de longueur 3 qui ne sont pas isolés (remarquons que $A_1 - A_3 - A_2 - A_1 - A_3 - A_4 - A_1$ n'est pas un circuit d'après notre définition),
- les deux circuits cités précédemment sont interconnectés (en A_1 et A_3),
- les chemins $D_1 - C_1 - B_1$ et $C_3 - B_2 - A$ sont indépendants, alors que $D_1 - C_1 - B_1 - A$ et $C_3 - B_2 - A$ sont racine-dépendants et que $D_1 - C_1 - B_1 - A$ et $C_2 - B_1 - A$ sont dépendants,
- D_1, C_2, E_1 sont les feuilles de $\mathcal{G}$,
- $D_1 - C_1 - B_1 - A$ est une branche d'attaque pour A, alors que $C_2 - B_1 - A$ est une branche de défense pour A,
- B_1 et B_2 sont les deux attaquants directs de A,
- C_1, C_2 et C_3 sont les trois défenseurs directs de A,
- D_1 et D_2 sont les deux attaquants indirects de A,
- E_1 est le seul défenseur indirect de A.

2.2 Acceptabilité collective

L'idée exploitée ici est que l'acceptabilité d'un argument dépend de son appartenance à certains ensembles (dits *ensembles acceptables* ou *extensions*) caractérisés par des propriétés telles que :

Définition 4 (Propriétés de base des extensions d'après [Dun95]) *Soit $<\mathcal{A}, \mathcal{R}>$ un système d'argumentation, on a :*

Ensemble sans conflit *Un ensemble $E \subseteq \mathcal{A}$ est* sans conflit *si et seulement si $\nexists A, B \in E$ tel que $A\mathcal{R}B$.*

Défense collective *Soit $E \subseteq \mathcal{A}$, $A \in \mathcal{A}$. E* défend (collectivement) *A si et seulement si $\forall B \in \mathcal{A}$, si $B\mathcal{R}A, \exists C \in E$ tel que $C\mathcal{R}B$. E* défend tous ses éléments *si et seulement si $\forall A \in E$, si $\exists B \in \mathcal{A}$ tel que $B\mathcal{R}A$ alors $\exists C \in E$ tel que $C\mathcal{R}B$.*

[Dun95] définit plusieurs sémantiques pour l'acceptabilité collective dont les *sémantiques admissible, préférée* et *stable* (avec, pour extensions respectives, les ensembles admissibles, les extensions préférées et les extensions stables) :

Définition 5 (Quelques extensions d'après [Dun95]) *Soit $<\mathcal{A}, \mathcal{R}>$ un système d'argumentation.*

[2] Les notions définies dans [Dun95] sont celles d'attaque et de défense indirecte, sachant que pour [Dun95] un attaquant (resp. défenseur) direct est aussi un attaquant (resp. défenseur) indirect, ce qui n'est pas le cas ici.

Ensemble admissible *Un ensemble $E \subseteq \mathcal{A}$ est* admissible *si et seulement si E est sans conflit et E défend tous ses éléments.*

Extension préférée *Un ensemble $E \subseteq \mathcal{A}$ est une* extension préférée *si et seulement si E est maximal pour l'inclusion parmi les ensembles admissibles.*

Extension stable *Un ensemble $E \subseteq \mathcal{A}$ est une* extension stable *si et seulement si E est sans conflit et E contrarie tout argument n'appartenant pas à E (c'est-à-dire $\forall A \in \mathcal{A} \setminus E$, $\exists B \in E$ tel que $B\mathcal{R}A$).*

Remarquons que l'acceptabilité d'un argument au sens de Dung est basée sur la prise en compte de chaque contrariant de cet argument pris séparément (il n'y a pas de notion de contrariété globale sur un argument). Nous rappelons aussi quelques propriétés essentielles :

Propriété 1 (Existence d'extension [Dun95]) *Soit $<\mathcal{A}, \mathcal{R}>$ un système d'argumentation, on a :*

1. *Tout ensemble admissible de $<\mathcal{A}, \mathcal{R}>$ est contenu dans une extension préférée de $<\mathcal{A}, \mathcal{R}>$.*
2. *$<\mathcal{A}, \mathcal{R}>$ possède au moins une extension préférée.*
3. *Si $<\mathcal{A}, \mathcal{R}>$ est bien-fondé alors il possède une et une seule extension préférée qui est aussi la seule extension stable.*
4. *Toute extension stable est aussi une extension préférée (et non vice-versa).*
5. *Il n'existe pas toujours d'extension stable.*

Propriété 2 *Soit E une extension préférée $\subseteq \mathcal{A}$, E contient tous les arguments non contrariés de $\mathcal{A}$.*

Preuve : Soit E une extension préférée $\subseteq \mathcal{A}$, supposons que E ne contienne pas tous les arguments non contrariés de $\mathcal{A}$. Donc, soit $A \in \mathcal{A}$ un argument non contrarié tel que $A \notin E$.
Étudions $E \cup \{A\}$:
- Soit $E \cup \{A\}$ est sans conflit et alors comme A est non contrarié et E une extension préférée, $E \cup \{A\}$ se défend collectivement, donc $E \cup \{A\}$ est admissible et $E \subseteq E \cup \{A\}$. Contradiction avec le fait que E est une extension préférée !
- Soit $E \cup \{A\}$ contient un conflit, donc :
  - Soit $\exists B \in E$ tel que $B\mathcal{R}A$, ce qui est impossible puisque A est non contrarié.
  - Soit $\exists B \in E$ tel que $A\mathcal{R}B$. Or, comme A est non contrarié, $\nexists C \in E$ tel que $C\mathcal{R}A$. Donc, E ne peut pas défendre B collectivement, ce qui est en contradiction avec E extension préférée !

En conclusion, l'hypothèse "E ne contient pas tous les arguments non contrariés de $\mathcal{A}$" débouche uniquement sur des contradictions. Elle est donc fausse. □

Propriété 3 *Soit E une extension stable $\subseteq \mathcal{A}$, E contient tous les arguments non contrariés de $\mathcal{A}$.*

Preuve : Soit E une extension stable $\subseteq \mathcal{A}$, supposons que E ne contienne pas tous les arguments non contrariés de $\mathcal{A}$. Donc, soit $A \in \mathcal{A}$ un argument non contrarié tel que $A \notin E$.
Puisque $A \notin E$ alors il doit exister dans E un autre argument B qui contrarie A ; or, cela est impossible puisque A est non contrarié.
Donc, l'hypothèse "E ne contient pas tous les arguments non contrariés de $\mathcal{A}$" débouche sur une contradiction. Elle est donc fausse. □

2.3 Différents niveaux d'acceptabilité d'un argument

2.3.1 Niveaux de base

Sous une sémantique donnée, l'acceptabilité d'un argument dépend, d'après Dung, de son appartenance à une extension de cette sémantique. Cela conduit à trois états possibles :
- soit l'argument est *uni-accepté*, car il appartient à toutes les extensions pour cette sémantique,
- soit l'argument est *exi-accepté*, car il appartient à au moins une extension pour cette sémantique,
- soit il est *non-accepté* car il n'appartient à aucune extension pour cette sémantique.

Toutefois, ces 3 niveaux d'acceptabilité nous paraissent insuffisants. Par exemple, que conclure dans le cas d'arguments se contrariant l'un l'autre et qui seraient tous deux exi-acceptés ?

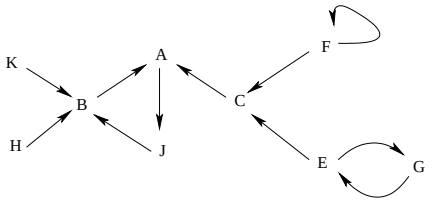
2.3.2 Premier raffinement des niveaux de base

On propose alors une nouvelle définition, celle d'un argument *proprement-accepté* qui tient compte de la situation d'un argument par rapport à ses contrariants pour une sémantique donnée S et ainsi permet de raffiner la classe des arguments exi-acceptés.

Définition 6 *Soit $A \in \mathcal{A}$, A est* proprement-accepté pour S *ssi A appartient à une extension pour S et $\forall B \in \mathcal{A}$ tel que $B\mathcal{R}A$, B n'appartient pas à une extension pour S.*

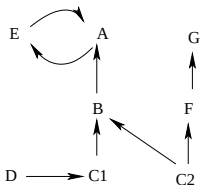
Ainsi, on essaye de capturer l'idée qu'un argument sera meilleur du point de vue de l'acceptabilité si ses attaquants directs ne sont pas acceptés.

Propriété 4 *Soit S une sémantique au sens de [Dun95]. Soit $A \in \mathcal{A}$.*
Si A est uni-accepté alors A est proprement-accepté. La réciproque est fausse, en général.

> **Preuve :** Un argument et un contrariant ne peuvent appartenir à une même extension au sens de [Dun95] car elle est sans conflit. Donc, puisque A est uni-accepté cela signifie que A appartient à toutes les extensions, donc aucun des attaquants directs de A ne peut appartenir à un de ces ensembles.
>
> Pour la réciproque, utilisons le contre-exemple suivant dans le cadre de la sémantique préférée :
>
> 
>
>
> On a ici 2 extensions préférées $\{K, H, G\}$ et $\{A, E, K, H\}$. L'argument A est proprement-accepté (B et C n'appartenant à aucune extension préférée, alors que A appartient à une de deux extensions). Par contre, A n'est pas uni-accepté puisqu'il n'appartient pas à toutes les extensions préférées.
>
> □

Cette notion permet de raffiner les niveaux déjà vus. Pour une sémantique S et un argument A, on a les différents états :
- A est *uni-accepté*, si $A \in$ à toutes les extensions pour S (il sera donc aussi proprement-accepté),
- A est *proprement-accepté* (donc par définition exi-accepté),
- A est *seulement-exi-accepté*, si A n'est pas proprement-accepté mais A exi-accepté.
- A est *non-accepté* si $A \notin$ à aucune extension pour S (donc, par définition, il ne pourra pas être proprement-accepté !).

Un exemple Soit le système d'argumentation suivant



On a deux extensions préférées $\{D, C_2, A, G\}$ et $\{D, C_2, E, G\}$. Donc, pour la sémantique préférée, les niveaux d'acceptabilité sont les suivants :
- D, C_2 et G sont uni-acceptés,
- il n'y a pas d'argument proprement-accepté,
- A et E sont seulement-exi-acceptés,
- B, C_1 et F sont non-acceptés.

2.3.3 Quelques cas particuliers

Dans tous les cas où il n'existe qu'une seule extension, les trois premières classes se confondent. Par exemple :
- Sous la sémantique préférée, quand il n'y a pas de circuit pair (résultat issu de [Dou02]).
- Sous la sémantique de base (c'est une autre sémantique proposée par Dung – voir [Dun95, Dou02] – et non présentée ici qui présente la particularité de n'avoir qu'une seule extension).

En effet, s'il n'existe qu'une seule extension, alors le fait que A appartienne à toutes les extensions est équivalent au fait qu'il appartienne au moins à une des extensions. D'autre part, avec une seule extension contenant A, les contrariants de A ne sont donc dans aucune extension et A est alors proprement accepté. Mais on a aussi :
- Sous la sémantique stable, la classe des uni-acceptés se confond avec celle des proprement-acceptés :

> Montrons d'abord que A uni-accepté $\Rightarrow$ A proprement-accepté : $\forall E$, E extension stable, $A \in E$; E extension stable donc $\forall B \notin E$, $\exists C \in E$ tel que $C\mathcal{R}B$, en particulier, pour les contrariants de A puisque A appartient à toutes les E extensions stables ; donc chacun des contrariants de A

est attaqué par chaque extension stable ; donc aucun des contrariants de A ne peut appartenir à une extension stable ; donc A est proprement-accepté.

Dans l'autre sens, soit A proprement-accepté, donc il existe E extension stable telle que $A \in E$ et $\forall B, B\mathcal{R}A$, $B \notin E'$, $\forall E'$ extension stable. Raisonnons par l'absurde en supposant qu'il existe une extension stable E'' telle que $A \notin E''$; or, si $A \notin E''$, cela implique que $\exists B \in E''$ tel que $B\mathcal{R}A$, donc B contrariant de A appartient à une extension stable ; donc contradiction avec l'hypothèse A proprement-accepté ; donc E'' n'existe pas et A est uni-accepté.

– Sous la sémantique préférée, quand il n'y a pas de circuit impair, la classe des uni-acceptés se confond avec celle des proprement-acceptés :

S'il n'existe pas de circuit impair, alors on peut exploiter un résultat issu de [DBC01, DBC02] repris dans [Dou02] qui montre que, dans ce cas-là, toute extension préférée est stable (ce qui correspond à la notion de système d'argumentation cohérent de [Dun95]). Ainsi, on réutilise le résultat précédent concernant la sémantique stable pour l'appliquer à la sémantique préférée.

Chapitre 3

Évaluation graduelle des interactions

Nous avons proposé deux méthodes d'évaluation distinctes dont l'objectif est de tenir compte de la qualité des attaquants et des défenseurs d'un argument.

3.1 L'approche "locale" de [CLS01] (évaluation générique)

Cette approche propose de calculer la valeur d'un argument uniquement en fonction des valeurs de ses contrariants. Il s'agit d'une approche générique qui recouvre des travaux existants ([BH01, JV99]).
On formalise cette évaluation en prenant l'hypothèse suivante : on dispose d'un ensemble W totalement ordonné par une relation notée $\geq$ et admettant un plus petit élément (V_{Min}) et d'une partie V de W, contenant V_{Min} et admettant un plus grand élément V_{Max}.

Définition 7 (Évaluation graduelle générique) *Soit $<\mathcal{A}, \mathcal{R}>$ un système d'argumentation. Une évaluation est une application $v : \mathcal{A} \to V$ telle que :*

1. $\forall A \in \mathcal{A}$, $v(A) \geq V_{Min}$
2. $\forall A \in \mathcal{A}$, *si* $\mathcal{R}^-(A) = \varnothing$, *alors* $v(A) = V_{Max}$
3. $\forall A \in \mathcal{A}$, *si* $\mathcal{R}^-(A) = \{A_1, \ldots, A_n\} \neq \varnothing$, *alors* $v(A) = g(h(v(A_1), \ldots, v(A_n)))$

où $h : V^* \to W$ telle que (V^* dénote l'ensemble des suites finies d'éléments de V)
– $h(x) = x$
– $h() = V_{Min}$
– $h(x_1, \ldots, x_n, x_{n+1}) \geq h(x_1, \ldots, x_n)$
et $g : W \to V$ telle que
– $g(V_{Min}) = V_{Max}$
– $g(V_{Max}) < V_{Max}$
– *g est decroissante (si $x \leq y$ alors $g(x) \geq g(y)$)*

Remarquons que $h(x_1, \ldots, x_n) \geq max(x_1, \ldots, x_n)$ est une conséquence des propriétés de la fonction h. Nous avons ici quelques propriétés dont certaines sont issues de [CLS01] :

Propriété 5 (voir [CLS01]) *La fonction g satisfait pour tout $n \geq 1$:*

$$g(V_{Max}) \leq g^3(V_{Max}) \leq \ldots \leq g^{2n+1}(V_{Max}) \leq g^{2n}(V_{Max}) \leq \ldots \leq g^2(V_{Max}) \leq V_{Max}$$

Si de plus g est strictement décroissante et $g(V_{Max}) > V_{Min}$, les inégalités ci-dessus sont strictes.

Propriété 6 (Pré-ordre total) *Une évaluation v donnée par la définition 7 induit un pré-ordre total $\succeq$ sur l'ensemble des arguments $\mathcal{A}$ défini par : $A \succeq B \Leftrightarrow v(A) \geq v(B)$.*

> **Preuve :** La fonction d'évaluation v associe à chaque argument A une valeur $v(A)$ dans un ensemble V qui est une partie d'un ensemble W totalement ordonné. □

Propriété 7 (Valeur dans un circuit) *Soit un circuit isolé de longueur n dans le graphe des contrariétés. Si n est impair, tous les arguments du circuit ont la même valeur qui est un point fixe de g, et si n est pair, chaque argument du circuit a pour valeur un point fixe de g^n.*

Preuve : Soit le circuit $\mathcal{C} = A_n - A_{n-1} - \ldots - A_2 - A_1$, on va travailler suivant la parité de n :
- Si n est pair : $n = 2k$ et $v(A_1) = g(v(A_2)) = \ldots = g^{2k-1}(v(A_{2k})) = g^{2k}(v(A_1))$, donc $v(A_1)$ est un point fixe de $g^{2k} = g^n$. De même pour tout $1 \leq i \leq 2k$.
  Par contre, les A_i n'ont pas forcément la même valeur : par exemple, pour $n = 2$, avec l'évaluation de [JV99], on a $v(A_1) = +$ et $v(A_2) = -$ avec $g(+) = -$ et $g(-) = +$. Si on impose que tous les A_i ont la même valeur, alors cette valeur sera un point fixe de g (puisque on aura $v(A_1) = g(v(A_2)) = g(v(A_1))$!).
- Si n est impair : $n = 2k + 1$ et $v(A_1) = g(v(A_2)) = \ldots = g^{2k}(v(A_{2k+1})) = g^{2k+1}(v(A_1))$, donc $v(A_1)$ est un point fixe de $g^{2k+1} = g^n$. De même pour tout $1 \leq i \leq 2k + 1$.
  Or, comme la fonction g est décroissante, la fonction g^{2k+1} l'est aussi et on a alors le résultat mathématique suivant qui s'applique : "si une fonction décroissante admet des points fixes, ils sont tous égaux"[1]. Ainsi, $v(A_1) = \ldots = v(A_{2k+1})$. Or, $v(A_1) = g(v(A_2)) = g(v(A_1))$, donc $v(A_1)$ est un point fixe de g.
  Ainsi, pour tout $1 \leq i \leq 2k + 1$, $v(A_i)$ est un point fixe de g.

□

Propriété 8 (Principes respectés – voir [CLS01]) *L'évaluation graduelle donnée par la définition 7 page précédente respecte les principes suivants :*

P1 *L'évaluation est maximale pour un argument sans attaquant et non maximale pour un argument attaqué et non défendu.*

P2 *L'évaluation d'un argument est fonction de l'évaluation de tous les attaquants directs (la contrariété directe).*

P3 *L'évaluation d'un argument est une fonction décroissante de l'évaluation de la contrariété directe.*

P4 *La contribution d'un attaquant direct d'un argument ne peut pas diminuer l'évaluation de la contrariété directe pour cet argument.*

La définition 7 page précédente caractérise une classe d'évaluations graduelles dont certaines instances sont déjà connues :

Propriété 9 (Lien avec [JV99] – voir [CLS01]) *Tout étiquetage enraciné de $<\mathcal{A}, \mathcal{R}>$ au sens de [JV99] peut être défini comme une instance v de l'évaluation générique telle que :*
- $V = W = \{-, ?, +\}$ *avec* $- < ? < +$,
- V_{Min} *dénoté par* $-$,
- V_{Max} *dénoté par* $+$,
- g *définie par* $g(-) = +$, $g(+) = -$, $g(?) = ?$
- *et* h *étant la fonction max.*

Propriété 10 (Lien avec [BH01] – voir [CLS01]) *La fonction d'évaluation graduelle de [BH01] peut être définie comme une instance v de l'évaluation générique telle que :*
- $V = [0, 1]$,
- $W = [0, \infty[$,
- $V_{Min} = 0$,
- $V_{Max} = 1$,
- $g : W \to V$ *définie par* $g(x) = \frac{1}{1+x}$
- *et* h *par* $h(\{x_1, \ldots, x_n\}) = x_1 + \ldots + x_n$.

3.2 L'approche "globale" de [CLS01] (évaluation par tuples)

Cette approche consiste à étiqueter un argument avec un tuple (on parlera alors de la valeur tuplée de l'argument), puis à comparer les arguments à l'aide de ces valeur tuplées.

[1] La preuve de ce résultat est très simple : soit g fonction décroissante, soient α et β deux points fixes de g. Si $\alpha \neq \beta$, on peut supposer que $\alpha > \beta$, donc que $g(\alpha) \leq g(\beta)$ (par la décroissance de g), donc que $\alpha \leq \beta$ (puisque α et β sont des points fixes de g) ce qui est une contradiction avec l'hypothèse $\alpha > \beta$. Choisir l'hypothèse inverse mène aussi à une contradiction. Donc $\alpha \neq \beta$ est fausse.

3.2.1 L'étiquetage d'un argument par un tuple

L'idée est d'avoir une étiquette qui reflète la structure de la partie du graphe des contrariétés dont l'argument étiqueté est la racine. L'étiquette pour un argument A est donc un tuple répertoriant les longueurs de toutes les branches d'attaque et de toutes les branches de défense menant à A. Le cas d'un graphe des contrariétés comportant des circuits fait l'objet de règles spécifiques dans lesquelles on considère le circuit comme un méta-argument possédant deux branches, une d'attaque de longueur 1 et une de défense de longueur 2 (chaque argument du circuit est à la fois attaqué et défendu par le circuit) sur lequel se rajoute ensuite l'impact des attaquants extérieurs au circuit et celui des autres circuits interconnectés. Toutes ces idées ont débouché dans [CLS01] sur les règles suivantes :

Règle 1 (Valeur d'un argument n'appartenant pas à un circuit) *Soit l'argument A n'appartenant pas à un circuit, sa valeur sera :*

si A n'est pas attaqué	$v(A) = ()$ *(aussi notée : $v(A) = (0)$)*
si A a pour attaquants directs les arguments $B_1, \ldots, B_n$ dont les valeurs respectives sont les tuples $(b_1^1, \ldots, b_{m_1}^1), \ldots, (b_1^n, \ldots, b_{m_n}^n)$	$v(A) = (b_1^1 + 1, \ldots, b_{m_1}^1 + 1, \ldots, b_1^n + 1, \ldots, b_{m_n}^n + 1)$

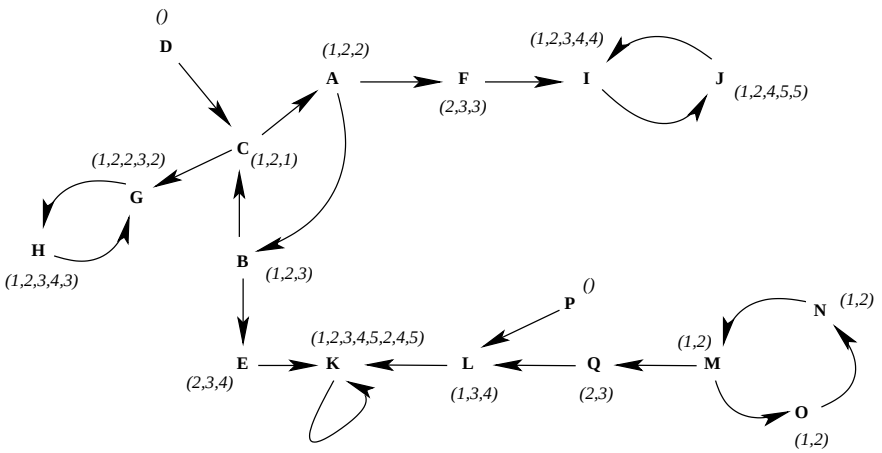
Règle 2 (Valeur d'un argument appartenant à un circuit) *Soit un argument A appartenant aux circuits $\mathtt{C}_1$ ... $\mathtt{C}_m$. Soit $\mathtt{C}'_1$ ... $\mathtt{C}'_n$ autres circuits interconnectés à l'un des $\mathtt{C}_i$ ou entre eux[2]. Soit X^1 ... X^p arguments n'appartenant pas aux circuits mais attaquants directs de l'un des circuits[3]. On note :*
- *l_i : longueur minimale d'un chemin d'un élément de $\mathtt{C}'_i$ vers A,*
- *l_{X^i} (longueur[4] d'un chemin de X^i vers A),*
- *chaque argument X^i a pour valeur : $v(X^i) = (x_1^i, \ldots, x_{k^i}^i)$.*

on a alors :

$$
\begin{aligned}
v(A) = (&\overbrace{1, 2, \ldots, 1, 2,}^{m \text{ fois}} \\
&1 + l_1, 2 + l_1, \\
&\ldots, \\
&1 + l_n, 2 + l_n, \\
&x_1^1 + l_{X^1}, \ldots, x_{k^1}^1 + l_{X^1}, \\
&\ldots, \\
&x_1^i + l_{X^i}, \ldots, x_{k^i}^i + l_{X_i}, \qquad \text{autant de fois qu'il y a de chemins de } X^i \text{ vers } A \\
&\ldots, \\
&x_1^p + l_{X^p}, \ldots, x_{k^p}^p + l_{X_p})
\end{aligned}
$$

Des exemples d'étiquetage complet : Prenons le graphe des contrariétés suivant et appliquons les règles précédentes :

[2] A n'appartient donc à aucun des $\mathtt{C}'_i$.
[3] C'est-à-dire attaquants directs d'éléments des circuits : $\forall X^i$, $\exists \mathtt{C} \in \{\mathtt{C}_1, \ldots, \mathtt{C}_m, \mathtt{C}'_1, \ldots, \mathtt{C}'_n\}$ tel que $\exists Y \in \mathtt{C}$ et $X^i \mathcal{R} Y$.
[4] S'il y a plusieurs chemins menant de X^i à A, on prendra en compte bien-sûr les longueurs de chacun de ces chemins.



Pour D, E, F, L, Q et P, on utilise la règle 1 page précédente. Les valeurs de tous les autres arguments sont fournies par la règle 2 page précédente.

Définition 8 (Évaluation graduelle par tuples) *Soit v une application de $\mathcal{A}$ dans l'ensemble des suites d'entiers naturels, v sera une évaluation par tuples si et seulement si v respecte les règles 1 page précédente et 2 page précédente.*

3.2.2 Comparaison d'arguments à l'aide de tuples

3.2.2.1 Les idées utilisées

Étant donné que, dans un tuple, les valeurs paires (correspondant aux branches de défense) et les valeurs impaires (correspondant aux branches d'attaque) ne jouent pas les mêmes rôles, nous ne pouvons pas nous contenter d'une simple comparaison lexicographique. Nous avons donc choisi de comparer les tuples en deux phases bien distinctes en adoptant une *attitude prudente* :

- Une "première passe" permet de déterminer et *comparer le nombre de branches d'attaque et le nombre de branches de défense* de chacun. Cela nous donne deux critères (un pour la défense et un pour l'attaque). Si ces deux critères sont en accord, c'est-à-dire que l'un des tuples a plus de branches de défense et moins de branches d'attaque que l'autre, alors on peut conclure. De même, si les deux critères sont en désaccord (l'un des tuples a plus de branches de défense et plus de branches d'attaque que l'autre), les tuples sont considérés comme incomparables.
- Sinon, les deux tuples ayant le même nombre d'attaques et le même nombre de défenses, une "seconde passe" *compare la qualité respective des attaques et des défenses*. La comparaison porte d'une part sur les "parties paires" des tuples, d'autre part sur les "parties impaires" des tuples. Ici-aussi, en cas de désaccord, la conclusion est l'incomparabilité des tuples ! Cette comparaison utilise un principe lexicographique.

Voici quelques exemples pour fixer les idées :

- $(1,2)$ est meilleur que $(1,1,2)$ car on a moins de branches d'attaque dans le premier tuple que dans le second pour un nombre de branches de défense identique (première passe).
- $(1,2)$ est incomparable avec $(1,1,2,2)$ car on a, dans le premier tuple, à la fois moins de branches d'attaque et moins de branches de défense que dans le second (première passe).
- $(2,3)$ est meilleur que $(1,2)$ car on a, dans le premier tuple, de meilleures branches d'attaque que dans le second (elles sont plus longues) pour des branches de défense identiques (seconde passe).
- $(2,3)$ est meilleur que $(3,4)$ car on a, dans le premier tuple, de meilleures branches de défense que dans le second (elles sont plus courtes) pour des branches d'attaque identiques (seconde passe).
- $(1,2)$ est incomparable avec $(3,4)$ car on a, dans le premier tuple, à la fois de plus mauvaises branches d'attaque et de meilleures branches de défense que dans le second (seconde passe).

Toutes ces idées ont débouchées sur l'algorithme 1 donné dans la section 3.2.2.2.

3.2.2.2 L'algorithme de comparaison

Notations Soit X un argument et $v(X)$ sa valeur, posons :

- $n_a(X)$ = nombre de branches d'attaque menant à X (c'est-à-dire nombre d'éléments impairs dans le tuple $v(X)$), en posant que $n_a(X) = 0$ si X est une feuille ;
- $n_d(X)$ = nombre de branches de défense menant à X (c'est-à-dire nombre d'éléments pairs dans le tuple $v(X)$), en posant que $n_d(X) = \infty$ si X est une feuille ;
- $v'(X) = f_c(v(X))$, la fonction $f_c(t)$ consistant à ordonner le tuple t par valeurs croissantes,

– $v'_p(X) = f_p(v'(X))$, la fonction $f_p(t)$ consistant à ne garder du tuple t que les valeurs paires (celles représentant des longueurs de branches de défense),
– $v'_i(X) = f_i(v'(X))$, la fonction $f_i(t)$ consistant à ne garder du tuple t que les valeurs impaires (celles représentant des longueurs de branches d'attaque).

Justification des choix Les données $n_a(X)$, $n_d(X)$ servent à effectuer la comparaison *quantitative* (pour savoir si X est meilleur que Y, il faut que X ait moins d'attaques ET plus de défenses que Y).
Quant à $v'(X)$, $v'_p(X)$, $v'_i(X)$, ils permettront la comparaison lexicographique des tuples dans le cas où X et Y auraient le même nombre d'attaques et de défenses. Remarquons que pour X dont la valeur est (), on a $v'(X) = v'_p(X) = v'_i(X) = ()$.

Remarque : Il est impossible pour un argument A d'avoir $n_a(A) = n_d(A) = 0$ (en effet, $n_a(A) = 0$ implique qu'aucune branche d'attaque n'arrive à A et $n_d(A) = 0$ implique qu'aucune branche de défense n'arrive à A, donc, puisqu'aucune branche – quelle que soit sa nature – n'arrive sur A, A est une feuille ; or, si A est une feuille, on doit avoir $n_d(A) = \infty$! Contradiction.).

Définition La comparaison[5] de 2 arguments A et B va se faire d'après l'algorithme 1.

Algorithme 1: Comparaison de 2 tuples en utilisant les comparaisons des nombres de branches

```
% Description des paramètres :                                              %
% v(A), v(B) : 2 tuples                                                     %
début
  si v(A) = v(B) alors A ≽ B ET B ≽ A                                  % Cas 1 %
  sinon
    si n_a(A) = n_a(B) et n_d(A) = n_d(B) alors
                % comparaisons lexicographiques entre v'_p(A) et v'_p(B) et entre v'_i(A) et v'_i(B) %
      si v'_p(A) ≤_lex v'_p(B) ET v'_i(A) ≥_lex v'_i(B) alors A ≻ B        % cas 2 %
      sinon
        si v'_p(A) ≥_lex v'_p(B) ET v'_i(A) ≤_lex v'_i(B) alors A ≺ B      % cas 3 %
        sinon A ⊁ B et A ⊀ B          % On décide de ne pas trancher ! cas 4 %
    sinon
      si n_a(A) ≥ n_a(B) ET n_d(A) ≤ n_d(B) alors A ≺ B                    % cas 5 %
      sinon
        si n_a(A) ≤ n_a(B) ET n_d(A) ≥ n_d(B) alors A ≻ B                  % cas 6 %
        sinon A ⊁ B et A ⊀ B          % On décide de ne pas trancher ! Cas 7 %
fin
```

3.2.3 Quelques propriétés

Soit v une évaluation par tuples, on a les propriétés suivantes dont certaines sont issues de [CLS01] :

Propriété 11 (Pré-ordre partiel et Valeur minimale) *La fonction d'évaluation par tuples associée à l'algorithme 1 induit un pré-ordre partiel $\succeq$ sur l'ensemble des arguments dont les éléments maximaux seront les feuilles ayant pour valeur ().*
L'argument $\perp$ ayant pour valeur le tuple $\underbrace{(1,\ldots,1)}_{\infty}$ est la valeur minimale[6] pour la relation $\succeq$.

Preuve : La preuve pour le pré-ordre est donnée dans [CLS01].
Pour la valeur minimale, cela découle directement de l'algorithme 1. En effet, il suffit de constater que tout argument ayant au moins une branche de défense sera toujours meilleur que $\perp$ (cas 5 et 6 de l'algorithme de comparaison), et que tout argument différent de $\perp$ n'ayant que

[5] Dans [CLS01], nous avions proposé deux méthodes de comparaison. Ici, nous n'en reprendrons qu'une seule, celle qui induit un pré-ordre partiel sur l'ensemble des arguments.
[6] Remarquons que cet argument correspond au cas d'un argument contrarié par une infinité d'arguments eux-mêmes jamais attaqués.

des branches d'attaque sera toujours meilleur que $\perp$ (cas 5 et 6, si le nombre de branches est différent, cas 2 et 3 si le nombre de branches est identique). □

Propriété 12 (Indépendance des branches) *Soit un argument A ayant pour attaquants directs A_1 de valeur $v(A_1) = (a^1_1, \ldots, a^1_{m_1})$, ..., A_n de valeur $v(A_n) = (a^n_1, \ldots, a^n_{m_n})$. Soit l'argument A' ayant pour attaquants directs A^1_1 de valeur $v(A^1_1) = (a^1_1)$, ..., $A^1_{m_1}$ de valeur $v(A^1_{m_1}) = (a^1_{m_1})$, ..., A^n_1 de valeur $v(A^n_1) = (a^n_1)$, ..., $A^n_{m_n}$ de valeur $v(A^n_{m_n}) = (a^n_{m_n})$. Alors $v(A) = v(A')$.*

Preuve : Cela découle directement des règles de calcul 1 page 11 et 2 page 11. □

Cette propriété montre bien l'aspect "indépendant" de la prise en compte de chaque branche même quand les dites-branches ne sont pas indépendantes graphiquement ! Sur l'exemple suivant, A et A' ont exactement la même valeur $(2, 2)$ bien qu'ils soient racine de sous-graphes différents :

Les deux définitions qui suivent déterminent précisément les notions d'amélioration/détérioration des défenses/attaques composant les principes que respecte l'évaluation par tuples (voir propriété 13 page ci-contre).

Définition 9 (Amélioration/détérioration des défenses/attaques – partie 1) *Soit A un argument ayant pour valeur $v(A) \neq ()$ telle que $v'_p(A) = (x^p_1, \ldots, x^p_n)$ (le tuple des défenses, donc x^p_j est un entier pair, $\forall j \in [1..n]$) et $v'_i(A) = (x^i_1, \ldots, x^i_m)$ (le tuple des attaques, donc x^i_j est un entier impair, $\forall j \in [1..m]$). On définit :*

Une amélioration de la défense *consiste à*
- *soit rajouter une branche de défense à A :*
  $v'_p(A)$ devient $f_c(x^p_1, \ldots, x^p_n, x^p_{n+1})$ avec x^p_{n+1} entier pair ;
- *soit diminuer une branche de défense de A :*
  $\exists j \in [1..n]$ tel que $v'_p(A)$ devient $(x^p_1, \ldots, x^p_{j-1}, x'^p_j, x^p_{j+1}, \ldots, x^p_n)$ avec $x'^p_j < x^p_j$ et x'^p_j pair ;
- *soit supprimer la seule branche de défense menant à A :*
  si initialement $v(A) = (x^p_j)$ avec x^p_j pair et que $v(A)$ devient () ;

Une amélioration de l'attaque *consiste à*
- *soit rajouter une branche d'attaque à A :*
  $v'_i(A)$ devient $f_c(x^i_1, \ldots, x^i_m, x^i_{m+1})$ avec x^i_{m+1} entier impair ;
- *soit diminuer une branche d'attaque de A :*
  $\exists j \in [1..m]$ tel que $v'_i(A)$ devient $(x^i_1, \ldots, x^i_{j-1}, x'^i_j, x^i_{j+1}, \ldots, x^i_m)$ avec $x'^i_j < x^i_j$ et x'^i_j impair ;

Une détérioration de la défense *consiste à*
- *soit supprimer une branche de défense à A :*
  $\exists j \in [1..n]$ tel que $v'_p(A)$ devient $(x^p_1, \ldots, x^p_{j-1}, x^p_{j+1}, \ldots, x^p_n)$;
- *soit rallonger une branche de défense de A :*
  $\exists j \in [1..n]$ tel que $v'_p(A)$ devient $(x^p_1, \ldots, x^p_{j-1}, x'^p_j, x^p_{j+1}, \ldots, x^p_n)$ avec $x'^p_j > x^p_j$ et x'^p_j pair ;

Une détérioration de l'attaque *consiste à*
- *soit supprimer une branche d'attaque à A :*
  $\exists j \in [1..m]$ tel que $v'_i(A)$ devient $(x^i_1, \ldots, x^i_{j-1}, x^i_{j+1}, \ldots, x^i_m)$;
- *soit rallonger une branche d'attaque de A :*
  $\exists j \in [1..m]$ tel que $v'_i(A)$ devient $(x^i_1, \ldots, x^i_{j-1}, x'^i_j, x^i_{j+1}, \ldots, x^i_m)$ avec $x'^i_j > x^i_j$ et x'^i_j impair.

Définition 10 (Amélioration/détérioration des défenses/attaques – partie 2) *Soit A un argument ayant pour valeur $v(A) = ()$ (donc $v'_p(A) = v'_i(A) = ()$). On définit :*

Une amélioration de l'attaque *consiste à rajouter une branche d'attaque à A :*
$v'_i(A)$ devient (x^i_j) avec x^i_j entier impair ;

Une détérioration de la défense *consiste à rajouter une branche de défense à A :*
$v'_p(A)$ devient (x^p_j) avec x^p_j entier pair.

Propriété 13 (Principes respectés – voir [CLS01]) *Soit v une évaluation par tuples (donc définie par les règles 1 page 11 et 2 page 11) associée à l'algorithme 1 page 13, v respecte les principes suivants :*

P1′ *L'évaluation est maximale pour un argument sans attaquant et non maximale pour un argument attaqué (qu'il soit défendu ou pas).*

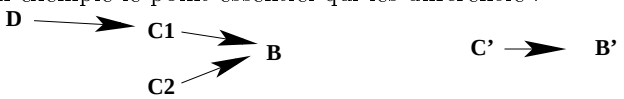
P2′ *L'évaluation d'un argument prend en compte toutes les branches dont cet argument est racine.*

P3′ *L'amélioration de la défense ou la détérioration de l'attaque conduisent à augmenter la valeur d'un argument.*

P4′ *L'amélioration de l'attaque ou la détérioration de la défense conduisent à diminuer la valeur d'un argument.*

3.3 Différences essentielles entre évaluations "locale" et "globale"

Dans [CLS01], le lecteur trouvera une comparaison de ces approches avec les autres approches du domaine ([Dun95, JV99, BH01]), ainsi qu'une comparaison approche "locale" – approche "globale".
Rappelons toutefois sur un exemple le point essentiel qui les différencie :



Sur cet exemple, avec l'approche locale, on constate que B a deux attaquants directs (C_2 de valeur maximale et C_1) alors que B' n'en a qu'un (C' de valeur maximale). Donc B' est meilleur que B.
Avec l'approche globale, on constate que deux branches (une d'attaque et une de défense) mènent à B alors qu'il y a une branche d'attaque qui mène à B'. Donc B est meilleur que B' (puisqu'il a au moins une défense alors que B' n'en a aucune). Dans ce cas-là, l'attaquant C_1 a perdu son statut négatif d'attaquant, puisque il est en fait "porteur d'une défense" pour B !
Ainsi, on peut synthétiser les résultats obtenus par ces approches sous la forme d'un tableau qui illustre bien les différences essentielles entre approche locale et approche globale lorsqu'on compare des arguments (seuls les arguments jamais attaqués sont partout les meilleurs !) :

Cas de l'approche globale

les arguments n'ayant que des branches d'attaque	$\preceq$	les arguments ayant des branches d'attaque et des branches de défense	$\preceq$	les arguments n'ayant que des branches de défense	$\preceq$	les arguments jamais attaqués

Cas de l'approche locale

les arguments ayant plusieurs attaquants directs non attaqués	$\preceq$	les arguments ayant un seul attaquant direct non attaqué	$\preceq$	les arguments ayant un seul attaquant direct attaqué (et éventuellement défendu)	$\preceq$	les arguments jamais attaqués
les arguments ayant plusieurs attaquants directs attaqués (et éventuellement défendus)						

Chapitre 4

Vers une acceptabilité graduelle

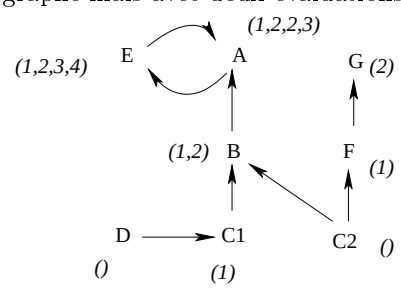
L'objectif est d'exploiter l'évaluation graduelle des interactions pour définir de nouveaux niveaux d'acceptabilité collective.

4.1 Comparaison basée sur l'évaluation graduelle

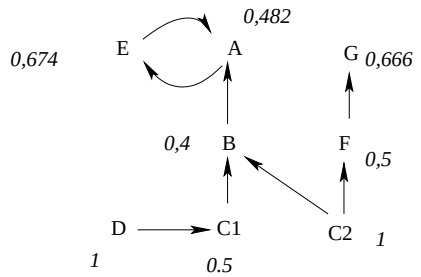
Soit v une évaluation graduelle, on peut définir une *v-préférence* utilisant le pré-ordre $\succeq$ issu de v :

Définition 11 *Soit $A, B \in \mathcal{A}$, A est* v-préféré à B *ssi $B \not\succ A$.*

Remarquons que, dans le cas où l'évaluation v induit un pré-ordre total sur l'ensemble des arguments, on a : A est *v-préféré à B* ssi $A \succeq B$.
Cette v-préférence peut être utilisée au sein d'un niveau d'acceptabilité (par exemple, celui des arguments exi-acceptés) pour obtenir des arguments mieux acceptés que d'autres. Attention, cette v-préférence est bien-sûr fonction de l'évaluation v choisie. Illustrons cette notion sur l'exemple suivant qui présente un même graphe mais avec deux évaluations graduelles différentes :



Évaluation par tuples



Évaluation de [BH01]

Avec l'évaluation par tuples, on obtient :

$$D, C_2 \succ G \succ \begin{matrix} B \\ A \succ E \end{matrix} \succ F, C_1 \qquad (B \text{ n'est pas comparable avec } A \text{ ou } E)$$

Avec l'évaluation de [BH01], on obtient :

$$D, C_2 \succ E \succ G \succ C_1, F \succ A \succ B$$

Si on applique cette v-préférence sans respecter les niveaux d'acceptabilité définis en section 2.3.2 page 6, on arrive à des situations contre-intuitives (par exemple, avec [BH01], E sera v-préféré à G alors que G est uni-accepté et E seulement-exi-accepté !).

4.2 Situation d'un argument par rapport à ses contrariants

Ici, on cherche à utiliser directement cette v-préférence pour situer, du point de vue de l'acceptabilité, un argument et ses contrariants.

Définition 12 *Soit $A \in \mathcal{A}$, A est* bien-défendu pour v *ssi* $\forall B \in \mathcal{A}$ *tel que* $B \mathcal{R} A$, A *est v-préféré à B.*

Ainsi, on essaye de capturer l'idée qu'un argument sera mieux accepté s'il est au moins aussi bon que ses attaquants directs (ou incomparable avec eux dans le cas d'un ordre partiel !). L'ensemble des arguments bien-défendus va dépendre de l'évaluation choisie. Si on reprend l'exemple donné en section 4.1 page précédente, on constate que les arguments bien-défendus sont D, C_2, G et A (A incomparable avec B mais meilleur que E) pour l'évaluation par tuples, alors qu'avec l'évaluation de [BH01] les arguments bien-défendus sont D, C_2, G et E (E meilleur que A).
Remarquons aussi que, comme dans le cas des sémantiques de [Dun95], la définition 12 considère les contrariants un par un. Elle n'est donc pas adaptée à une évaluation qui utiliserait la contrariété dans sa globalité (ce qui est par exemple le cas dans [BH01]).

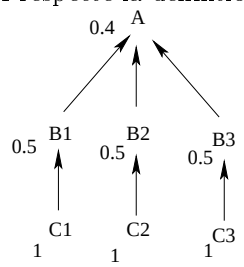
4.3 Compatibilité entre acceptabilité collective et évaluation graduelle

Nous disposons donc d'une nouvelle notion d'acceptabilité (définition 12) dont il faut étudier la compatibilité avec la partition en niveaux proposée en section 2.3.2 page 6.
On constate très vite sur divers exemples que ces deux approches ne sont pas compatibles.

4.3.1 Exemples montrant la non compatibilité dans le cas général

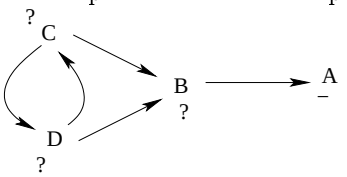
Passons en revue quelques exemples pour chacune des évaluations graduelles les plus connues (deux instances de l'évaluation locale générique : [BH01, JV99] et l'évaluation globale par tuples de [CLS01]) et pour les sémantiques de l'acceptabilité les plus classiques (les sémantiques préférée et stable de [Dun95]).

Argument proprement-accepté mais pas bien-défendu On a ici au moins 3 exemples :
– l'argument A respecte la définition 6 page 6 mais pas la définition 12 :



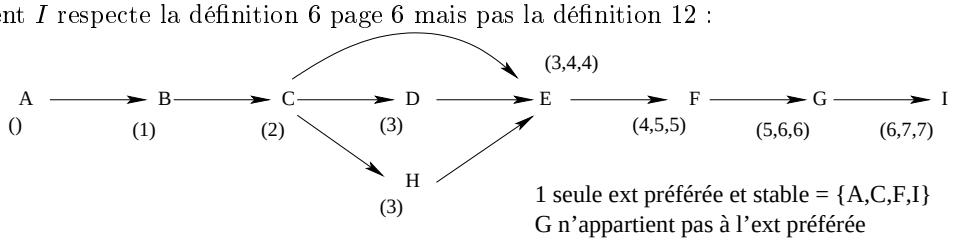
1 seule ext préférée et stable = { C1, C2, C3, A}
B1, B2, B3 n'appartiennent à aucune ext préférée
v(A) < v(Bi) quel que soit i = 1, 2, 3

– l'argument A respecte la définition 6 page 6 mais pas la définition 12 :



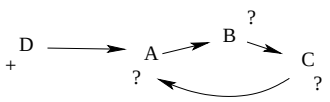
Ici, 2 ext préférées et stables : {C,A} et {D,A}
B n'appartient à aucune ext préférée
v(A) < v(B)

– l'argument I respecte la définition 6 page 6 mais pas la définition 12 :



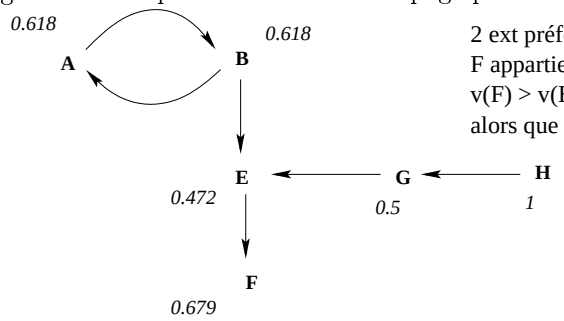
1 seule ext préférée et stable = {A,C,F,I}
G n'appartient pas à l'ext préférée
v(I) < v(G)

Argument bien-défendu mais pas proprement-accepté On a ici au moins 3 exemples :
– l'argument C respecte la définition 12 mais pas la définition 6 page 6 :



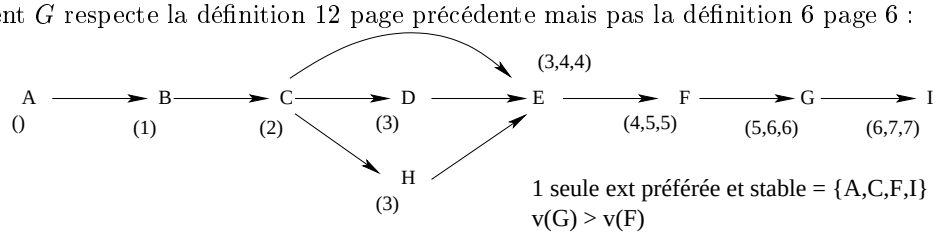
1 seule ext préférée et stable : {D, B}
v(C) >= v(B)
C n'appartient pas à l'ext préférée et stable

– l'argument F respecte la définition 12 page précédente mais pas la définition 6 page 6 :



2 ext préférées et stables = {A,H,E} et {B,H,F}
F appartient à 1 ext préférée et stable
v(F) > v(E) et E appartient aussi à 1 ext préférée et stable
alors que E attaque F

– l'argument G respecte la définition 12 page précédente mais pas la définition 6 page 6 :



1 seule ext préférée et stable = {A,C,F,I}
v(G) > v(F)
G n'appartient pas à l'ext préférée

4.3.2 Cas particuliers menant à la compatibilité

Nous nous plaçons dans le contexte d'un système d'argumentation avec une relation $\mathcal{R}$ finie et n'admettant aucun circuit[1]. Les sémantiques stable et préférée ne proposent alors qu'une seule extension et les niveaux d'acceptabilité uni-acceptés, exi-acceptés, proprement-acceptés fusionnent.
Dans ce contexte, il y a au moins deux cas particuliers menant à une compatibilité.

4.3.2.1 Premier cas

Il concerne l'évaluation graduelle globale par tuples :

Propriété 14 *Soit $\mathcal{G}$ le graphe associé à $<\mathcal{A},\mathcal{R}>$ un système d'argumentation avec $\mathcal{R}$ finie et n'admettant aucun circuit et vérifiant la condition suivante : $\exists\ A \in \mathcal{A}$ tel que*
– *$\forall X_i$, $\forall i = 1 \ldots n$, feuille de $\mathcal{G}$, $\exists$ un seul chemin de X_i vers A, $X_i^1 - \ldots - X_i^{l_i} - A$ avec $X_i^1 = X_i$ et l_i la longueur de ce chemin (si l_i est paire, ce chemin est une branche de défense pour A, sinon c'est une branche d'attaque),*
– *tous les chemins de X_i vers A sont racine-dépendants en A,*
– *$\forall A_i \in \mathcal{A}$, $\exists X_j$ une feuille telle que A_i appartienne à un chemin de X_j vers A.*
Soit v une évaluation par tuples. Soit S une sémantique $\in$ {préférée, stable}.
$\forall B \in \mathcal{A}$, B exi-accepté pour $S \Leftrightarrow B$ bien-défendu pour v.

Preuve : La preuve repose sur l'étude des contraintes entre valeurs sur chaque chemin. Soit X_i une feuille, le chemin $\mathcal{C} \in \mathcal{C}(X_i, A)$ est le suivant $X_i^1 - \ldots - X_i^{l_i} - A$ avec $X_i^1 = X_i$ et l_i la longueur de ce chemin (si l_i est paire, ce chemin est une branche de défense pour A, sinon c'est une branche d'attaque).
Les contraintes entre X_i^1 à $X_i^{l_i}$ sont les suivantes (voir [CLS01]) :

$$X_i^1 \succ X_i^3 \succ \ldots \succ X_i^{l_i} \succ X_i^2 \succ X_i^4 \succ \ldots \succ X_i^{l_i-1} \text{ si } l_i \text{ impaire } \geq 1$$

ou

$$X_i^1 \succ X_i^3 \succ \ldots \succ X_i^{l_i-1} \succ X_i^2 \succ X_i^4 \succ \ldots \succ X_i^{l_i} \text{ si } l_i \text{ paire } \geq 2$$

[1] Il s'agit donc d'un système d'argumentation $(\mathcal{A}, \mathcal{R})$ bien-fondé.

Donc, pour le chemin $X_i^1 - \ldots - X_i^{l_i}$, l'ensemble des arguments bien-défendus est $\{X_i^1, X_i^3, \ldots, X_i^{l_i}\}$ si l_i est impaire, $\{X_i^1, X_i^3, \ldots, X_i^{l_i-1}\}$ sinon (ce sont tous les arguments ayant une valeur strictement supérieure à celle de leurs attaquants directs et indirects). Nous noterons cet ensemble ACCEP_i.

Par définition, cet ensemble est sans-conflit, il défend ses éléments (puisqu'il ne contient que la feuille et **tous** les arguments du chemin défendus par cette feuille) et donc attaque tous les arguments restants. Si on cherche à rajouter n'importe quel argument $X \in \{X_i^1, \ldots, X_i^{l_i}\} \setminus \text{ACCEP}_i$, on obtient un conflit (puisque **tous** les arguments restants sont ceux qui sont contrariés par des éléments de ACCEP_i). Donc. pour $\{X_i^1, \ldots, X_i^{l_i}\}$, ACCEP_i est une extension préférée et stable.

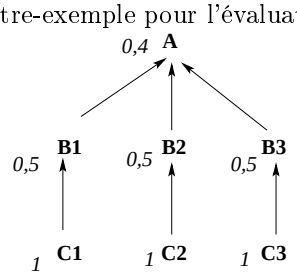
Posons $\mathcal{A}' = \mathcal{A} \setminus \{A\}$, $\mathcal{R}'$ la restriction de $\mathcal{R}$ à $\mathcal{A}'$[2] et $\text{UNION_ACCEP} = \cup_i \text{ACCEP}_i$, on a alors UNION_ACCEP qui est une extension préférée et stable de $(\mathcal{A}', \mathcal{R}')$.

Voyons maintenant le cas de A.

- Soit A n'est pas bien-défendu, donc il existe un attaquant direct $X_i^{l_i}$ de A tel que $v(A) < v(X_i^{l_i})$.
  On a deux cas :
  - soit $X_i^{l_i}$ correspond à l_i impaire donc $v(X_i^{l_i}) = (l_i - 1)$ (nombre pair !), ce qui signifie que $X_i^{l_i} \in \text{UNION_ACCEP}$. On a alors $\text{UNION_ACCEP} \cup \{A\}$ qui n'est pas sans conflit. Donc UNION_ACCEP est aussi une extension préférée et stable de $<\mathcal{A}, \mathcal{R}>$.
  - soit $X_i^{l_i}$ correspond à l_i paire donc $v(X_i^{l_i}) = (l_i - 1)$ (nombre impair !), donc pour que $v(A)$ soit inférieure à $v(X_i^{l_i})$ il faudrait que A n'ait aucune branche de défense ce qui est impossible puisque la branche venant de $X_i^{l_i}$ est une branche de défense pour A ! Donc contradiction. Ce cas ne peut pas se produire.
- Soit A est bien-défendu, ce qui signifie que **toutes** les branches menant à A sont des branches de défense pour A (donc l_i est paire pour chaque branche i). On a alors $\text{UNION_ACCEP} \cup \{A\}$ qui est sans conflit et A qui est défendu contre chacun de ses attaquants directs (puisque $X_i^{l_i-1} \in \text{UNION_ACCEP}$ pour chaque branche i). Donc $\text{UNION_ACCEP} \cup \{A\}$ est une extension préférée et stable de $<\mathcal{A}, \mathcal{R}>$.

On a donc l'ensemble des arguments bien-défendus qui est une extension préférée et stable de $<\mathcal{A}, \mathcal{R}>$. Et comme on est dans le cas où cette extension est unique, on peut donc conclure que $\forall B \in \mathcal{A}$, B exi-accepté $\Leftrightarrow$ B bien-défendu. □

Remarquons que cette propriété 14 page précédente n'est en général pas vérifiée par une évaluation générique. Ce schéma suivant fournit un contre-exemple pour l'évaluation de [BH01] :



Sur ce schéma, on constate que l'ensemble des arguments bien-défendus est $\{C_1, C_2, C_3\}$ (donc A pas-bien-défendu) et pourtant l'ensemble $\{C_1, C_2, C_3, A\}$ est un ensemble admissible (c'est même l'extension préférée).

4.3.2.2 Second cas

Ce second cas concerne l'évaluation graduelle locale générique :

Propriété 15 *Soit $<\mathcal{A}, \mathcal{R}>$ un système d'argumentation avec $\mathcal{R}$ finie et n'admettant aucun circuit. Soit S une sémantique $\in$ {préférée, stable}. Soit v une évaluation graduelle locale vérifiant la condition (i) suivante :*

$$(\forall i = 1 \ldots n, g(x_i) \geq x_i) \Rightarrow (g(h(x_1, \ldots, x_n)) \geq h(x_1, \ldots, x_n)) \qquad (i)$$

$\forall A \in \mathcal{A}$, A exi-accepté pour S $\Leftrightarrow$ A bien-défendu pour v.

[2] $\mathcal{R}'$ est la restriction de $\mathcal{R}$ à $\mathcal{A}'$ si et seulement si $\mathcal{R}' = \{(a, b) | a \mathcal{R} b, a \in \mathcal{A}', b \in \mathcal{A}'\}$.

Cette propriété utilise le lemme suivant :

Lemme 1 *Soit $<\mathcal{A}, \mathcal{R}>$ un système d'argumentation avec $\mathcal{R}$ finie et n'admettant aucun circuit. Soit S une sémantique $\in$ {préférée, stable}. Soit v une évaluation graduelle générique vérifiant la condition (i).*

1. *Cas 1 : Si A est exi-accepté et que A n'a qu'un seul contrariant noté B alors $v(A) \geq v(B)$.*
2. *Cas 2 : Si B est non-accepté et que B n'a qu'un seul contrariant noté C alors $v(B) \leq v(C)$.*

Preuve du lemme : Sachant que l'on est dans le contexte d'un système d'argumentation $<\mathcal{A}, \mathcal{R}>$ avec $\mathcal{R}$ finie et n'admettant aucun circuit (donc une seule extension préférée et stable notée E, donc non vide), on sait donc que :
- si A est exi-accepté et si A a un contrariant noté B alors B est non-accepté,
- si B est non-accepté alors il existe au moins un argument C tel que $C\mathcal{R}B$ et C exi-accepté (puisque B n'est pas dans E et que E est stable alors C doit $\in E$). Donc, *a fortiori*, si B est non-accepté et n'a qu'un seul contrariant C, alors C sera exi-accepté.

Puis, on fait la preuve par induction sur la profondeur d'un arbre de preuve pour A ou C.
- Cas de base pour le cas 1 : A exi-accepté avec un unique contrariant B, donc on a $B\mathcal{R}A$ et $C_1 \dots C_n$ contrariants de B, ce qui donne un arbre de preuve de longueur 2 pour A avec un des C_i non contrarié, par exemple C_1 ; on a alors :

$$\begin{array}{rcll} v(B) & = & g(h(v(C_1), \dots, v(C_n))) & \\ & \leq & g(v(C_1)) & \text{puisque } h(v(C_1), \dots, v(C_n)) \geq h(v(C_1)) = v(C_1) \\ & & & \text{et que } g \text{ est décroissante} \\ & \leq & g(V_{Max}) & \text{car } v(C_1) = V_{Max} \end{array}$$

  donc :

$$\begin{array}{rcl} v(A) & = & g(v(B)) \\ & \geq & g^2(V_{Max}) \end{array}$$

  Or, on sait par la propriété 5 page 9 que $g^2(V_{Max}) \geq g(V_{Max})$, donc $v(A) \geq v(B)$.
- Cas de base pour le cas 2 : on a $C\mathcal{R}B$ avec C unique contrariant de B, ce qui donne un arbre de preuve de longueur 0 pour C, c'est-à-dire C non contrarié ; donc $v(C) = V_{Max}$ et $v(B) = g(V_{Max}) \leq v(C)$ (d'après la définition 7 page 9).
- Cas général pour le cas 1 : A exi-accepté avec un unique contrariant B, donc on a $B\mathcal{R}A$ et $C_1 \dots C_n$ contrariants de B, avec un des C_i exi-accepté, par exemple C_1 ; considérons le sous-graphe menant à C_1 auquel on ajoute $C_1\mathcal{R}B\mathcal{R}A$, et posons l'hypothèse d'induction suivante (issue du cas 2) :

$$g(v(C_1)) \leq v(C_1)$$

  Donc :

$$\begin{array}{rcll} v(B) & = & g(h(v(C_1), \dots, v(C_n))) & \\ & \leq & g(v(C_1)) & \text{toujours pour les mêmes raisons que dans le cas de base} \\ & \leq & v(C_1) & \text{par hypothèse d'induction} \\ & \leq & h(v(C_1), \dots, v(C_n)) & \text{propriété de } h \end{array}$$

  on a donc, à cause de la décroissance de g :

$$\begin{array}{rcl} v(A) & = & g(v(B)) \\ & \geq & g(h(v(C_1), \dots, v(C_n))) = v(B) \end{array}$$

- Cas général pour le cas 2 : B est non-accepté, donc C est exi-accepté ; supposons que C admette plusieurs contrariants $D_1 \dots D_p$ qui sont tous non-acceptés (puisque C est exi-accepté) ;

considérons chaque sous-graphe menant à D_i auquel on ajoute $D_i\mathcal{R}C\mathcal{R}B$ et posons l'hypothèse d'induction suivante (issue du cas 1) :

$$\forall i = 1 \dots p,\ g(v(D_i)) \geq v(D_i)$$

donc :

$$\begin{array}{rcll} v(C) & = & g(h(v(D_1), \dots, v(D_p))) & \\ & \geq & h(v(D_1), \dots, v(D_p)) & \text{application de la condition } (i) \\ & & \multicolumn{2}{l}{\text{puisque l'hypothèse d'induction prise correspond à la prémisse de } (i)} \end{array}$$

donc :

$$\begin{array}{rcl} v(B) & = & g(v(C)) \\ & \leq & g(h(v(D_1), \dots, v(D_p))) = v(C) \end{array}$$

□

Preuve : Supposons que (i) vraie et étudions $A \in \mathcal{A}$ qui est exi-accepté. Soit B_i, $i = 1 \dots n$, les contrariants de A. Alors, pour chaque $i = 1 \dots n$, dans le sous-graphe menant à B_i et complété par $B_i\mathcal{R}A$, on applique le lemme et on a : $g(v(B_i)) \geq v(B_i)$, $\forall i = 1 \dots n$. Ainsi, on obtient :

$$\begin{array}{rcll} v(A) & = & g(h(v(B_1), \dots, v(B_n))) & \\ & \geq & h(v(B_1), \dots, v(B_n)) & \text{en appliquant } (i) \\ & \geq & v(B_i), \forall i = 1 \dots n & \text{propriété de } h \end{array}$$

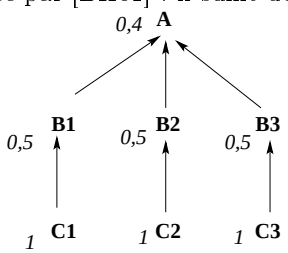
Donc, A est bien-défendu.

Pour la réciproque, soit $A \in \mathcal{A}$ bien-défendu. Notons $B_1, \dots, B_n$ les contrariants de A et supposons que A n'est pas exi-accepté. Alors il doit exister au moins un contrariant B_i de A tel que B_i est exi-accepté (puisqu'il n'y a qu'une seule extension préférée et stable). On peut donc appliquer le cas 2 du lemme sur le sous-graphe menant à B_i complété par $B_i\mathcal{R}A$ et on obtient $g(v(B_i)) \leq v(B_i)$. Donc, il existe B_i un contrariant de A tel que :

$$\begin{array}{rcll} v(A) & = & g(h(v(B_1), \dots, v(B_n))) & \\ & \leq & g(v(B_i)) & \text{propriété de } h \text{ et décroissance de } g \\ & \leq & v(B_i) & \text{utilisation du lemme} \end{array}$$

Ce qui est en contradiction avec A bien-défendu. Donc A est exi-accepté. □

Remarque : Cette condition (i) est :
– fausse pour l'évaluation locale proposée par [BH01] : il suffit de reprendre le graphe suivant :



Sachant que $g(x) = \frac{1}{1+x}$ et $h(x_1, \dots, x_n) = \Sigma_{i=1}^{n} x_i$ (voir propriété 10 page 10), on constate alors que :
– $\forall i = 1 \dots 3$, $x_i = v(B_i) = 0.5$,
– $\forall i = 1 \dots 3$, $g(x_i) = 0.66$, donc $g(x_i) \geq x_i$,
– et pourtant $g(h(x_1, x_2, x_3)) = v(A) = 0.4 \not\geq h(x_1, x_2, x_3) = 1.5$!

- vraie pour les évaluations locales définies avec $h = Max$ (quelle que soit la fonction g) : si $h = Max$ alors $g(h(x_1, \ldots, x_n)) = g(Max(x_1, \ldots, x_n)) = g(x_j)$, x_j étant le plus grand des x_i ; or, par hypothèse, $g(x_i) \geq x_i$, $\forall x_i$, donc en particulier pour x_j ; donc, on a :
$$g(h(x_1, \ldots, x_n)) = g(x_j) \geq x_j = Max(x_1, \ldots, x_n) = h(x_1, \ldots, x_n)$$
- fausse pour les évaluations locales définies avec h telle que $\exists n > 1$ avec $h(x_1, \ldots, x_n) > Max(x_1, \ldots, x_n)$ (quelle que soit la fonction g strictement décroissante) : il suffit de reprendre le contre-exemple proposé pour [BH01] en constatant que $h(x_1, x_2, x_3) = 1.5$ et que $Max(x_1, x_2, x_3) = 0.5$; donc, que l'on a $\exists n > 1 (= 3)$ avec $h(x_1, \ldots, x_n) > Max(x_1, \ldots, x_n)$.

Chapitre 5

Conclusion

Dans ce papier, nous avons cherché à gradualiser la notion d'acceptabilité d'un argument. Pour cela, nous nous sommes appuyées sur la notion d'acceptabilité collective de [Dun95] qui permet déjà d'avoir 3 niveaux d'acceptabilité (uni-acceptés, exi-acceptés, non-acceptés). Nous avons alors cherché à introduire la prise en compte d'une évaluation graduelle des arguments dans la notion d'acceptabilité collective.
Cela nous a amené à proposer diverses notions applicables à un argument :
- les arguments *proprement acceptés* : ceux dont les contrariants ne sont pas acceptés (pour une sémantique donnée),
- la *v-préférence* entre arguments issue directement de l'évaluation v,
- les arguments *bien-défendus* : ceux qui sont v-préférés à leurs contrariants (pour une évaluation graduelle v donnée).

La première notion permet de raffiner le niveau des exi-acceptés en deux sous-niveaux (les proprement-acceptés et les seulement-exi-acceptés).
La seconde notion permet de gradualiser la totalité des niveaux d'acceptabilité mais seulement en l'utilisant à l'intérieur de chaque niveau.
La troisième notion permet de définir deux nouveaux niveaux d'acceptabilité (les bien-défendus et les pas-bien-défendus). Malheureusement, nous avons constaté sur divers exemples que, dans le cas général, ces niveaux n'étaient pas compatibles avec ceux proposés par [Dun95] et raffinés à l'aide de la première notion (l'acceptabilité propre) ; et cela, bien que les notions d'argument bien-défendu et d'argument proprement-accepté soient toutes les deux construites à partir des interactions entre arguments.
Il reste quand même quelques cas particuliers sur lesquels on a une compatibilité que l'on peut exploiter.

Bibliographie

[AC98] Amgoud (Leila) et Cayrol (Claudette). – On the acceptability of arguments in preference-based argumentation. *In : Proc. of the 14th Uncertainty in Artificial Intelligence*, éd. par Cooper (G. F.) et Moral (S.). pp. 1–7. – Madison, Wisconsin, 1998.

[BH01] Besnard (Philippe) et Hunter (Anthony). – A logic-based theory of deductive arguments. *Artificial Intelligence*, vol. 128 (1-2), 2001, pp. 203–235.

[CLS01] Cayrol (Claudette) et Lagasquie-Schiex (Marie-Christine). – *Évaluation graduelle d'arguments.* – Rapport de recherche n2001-18-R, France, Institut de Recherche en Informatique de Toulouse (I.R.I.T.), novembre 2001.

[CLS02] Cayrol (Claudette) et Lagasquie-Schiex (Marie-Christine). – Gradual handling of contradiction in argumentation frameworks. *In : Proc. of the 9th IPMU (Information Processing and Management of Uncertainty in Knowledge-Based Systems)*, pp. 83–90. – Annecy, France, 2002.

[DBC01] Dunne (Paul E.) et Bench-Capon (T.J.M.). – *Coherence in finite argument systems.* – Technical report n01-006, University of Liverpool, Department of Computer Science (U.L.C.S.), 2001.

[DBC02] Dunne (Paul E.) et Bench-Capon (T.J.M.). – Coherence in finite argument system. *Artificial Intelligence*, vol. 141, n1-2, 2002, pp. 187–203.

[Dou02] Doutre (Sylvie). – *Autour de la sémantique préférée des systèmes d'argumentation.* – Thèse, Université Paul Sabatier, IRIT, 2002.

[Dun95] Dung (Phan Minh). – On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, vol. 77, 1995, pp. 321–357.

[JV99] Jakobovits (H.) et Vermeir (D.). – Robust semantics for argumentation frameworks. *Journal of logic and computation*, vol. 9(2), 1999, pp. 215–261.

[KAEF95] Krause (P.), Ambler (S.), Elvang (M.) et Fox (J.). – A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, vol. 11 (1), 1995, pp. 113–131.

[Par97] Parsons (S.). – Normative argumentation and qualitative probability. *LNAI*, vol. 1244, 1997, pp. 466–480.

[PS97] Prakken (H.) et Sartor (G.). – Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, vol. 7, 1997, pp. 25–75.