When Price is not enough: Combining Logical and Numerical Issues in Bilateral Negotiation*

Azzurra Ragone†, Tommaso Di Noia†, Eugenio Di Sciascio†, Francesco M. Donini‡
†SisInf Lab - Politecnico di Bari - Italy
{a.ragone, t.dinoia, disciascio}@poliba.it
‡SisInf Lab - Università della Tuscia - Italy
donini@unitus.it

ABSTRACT

We present a novel approach to knowledge-based automated one-shot multi-issue bilateral negotiation handling, in a homogeneous setting, both numerical features and non-numerical ones. To this aim we introduce $\mathcal{P}(\mathcal{N})$, a propositional logic extended with concrete domains, which allows to: model relations among issues (both numerical and not numerical ones) via logical entailment, differently from well-known approaches that describe issues as uncorrelated; represent buyer's request, seller's supply and their respective preferences as formulas endowed with a formal semantics.

Categories and Subject Descriptors: I.2.4 [Knowledge Representation Formalisms and Methods]

General Terms: Languages, Theory.

Keywords: negotiation, logic, propositional weighted formulas.

1. INTRODUCTION

In this work we focus on automated negotiation in e-marketplaces where it is not sufficient to deal with undifferentiated products (commodities as oil, cement, etc.) or stocks, taking into account only price, time or quantity but also other features have to be considered during the negotiation process, as warranty or delivery time, as well as – in a car marketplace – look, model, comfort and so on. To the best of our knowledge, also in recent literature, issues are usually described as uncorrelated terms, without considering any underlying semantics. In our approach we use knowledge representation in two ways: (1) exploiting an ontology to represent relations among issues and (2) assigning utilities to formulas to represent agents having preferences over different bundles of issues. Since issues are often inter-dependent, agents can express conditional preferences as "*if I spend more than 20000 € for a sedan then I want a navigator pack included*" where both numerical (price) and non-numerical (sedan, navigator pack) issues coexist and where the meaning of navigator pack is in the ontology ($\texttt{NavigatorPack} \Leftrightarrow \texttt{SatelliteAlarm} \wedge \texttt{GPS\_system}$) Contributions of this paper include: the framework for automated multi-issue bilateral negotiation; an extended propositional logic, $\mathcal{P}(\mathcal{N})$ enriched with concrete domains, which is able to handle both numerical features and not numerical ones as correlated issues w.r.t. an ontology, and the one-shot protocol we adopt, which allows to compute Pareto-efficient agreements, exploiting a mediator that solves a multi objective optimization problem.

2. REPRESENTATION OF ISSUES

We divide issues involved in a negotiation in two categories. Some issues may express properties that are true or false, like, *e.g.*, in an automotive domain, `ItalianMaker`, or `AlarmSystem`. We represent them as propositional atoms $A_1, A_2, \ldots$ from a finite set $\mathcal{A}$. Other issues involve numerical features like `deliverytime`, or `price` represented as variables $f_1, f_2, \ldots$, each one taking values in its specific domain $D_{f_1}, D_{f_2}, \ldots$, such as $[0, 90]$ (days) for `deliverytime`, or $[1,000, 20,000]$ (euros), for `price`. The variables representing numerical features are always constrained by comparing them to some constant, like $\texttt{price} < 20{,}000$, or $\texttt{deliverytime} \geq 30$, and such constraints can be combined into complex propositional requirements – also involving propositional issues – *e.g.*, $\texttt{ItalianMaker} \wedge (\texttt{price} \leq 25{,}000) \wedge (\texttt{year\_warr} \geq 3)$ (representing a car made in Italy, costing no more than 25,000 euros, with a warranty greater or equal than 3 years), or $\texttt{AlarmSystem} \Rightarrow (\texttt{deliverytime} > 30)$ (expressing the seller's requirement "if you want an alarm system mounted you'll have to wait more than one month"). We define:

DEFINITION 1 (CONCRETE DOMAINS, [1]). *A concrete domain D consists of a finite set $\Delta_c(D)$ of numerical values, and a set of predicates $C(D)$ expressing numerical constraints on D.*

For our numerical features, predicates will always be the binary operators $C(D) = \{\geq, \leq, >, <, =, \neq\}$, whose second argument is a constant in $\Delta_c(D)$. Now we can formally extend propositional logic in order to handle numerical features. We call this language $\mathcal{P}(\mathcal{N})$.

DEFINITION 2 (THE LANGUAGE $\mathcal{P}(\mathcal{N})$). *Let $\mathcal{A}$ be a set of propositional atoms, and F a set of pairs $\langle f, D_f \rangle$ each made of a feature name and an associated concrete domain D_f, and let k be a value in D_f. Then the following formulas are in $\mathcal{P}(\mathcal{N})$:*
1. *every atom $A \in \mathcal{A}$ is a formula in $\mathcal{P}(\mathcal{N})$*
2. *if $\langle f, D_f \rangle \in F$, $k \in D_f$, and $c \in \{\geq, \leq, >, <, =, \neq\}$ then (fck) is a formula in $\mathcal{P}(\mathcal{N})$*
3. *if ψ and φ are formulas in $\mathcal{P}(\mathcal{N})$ then $\neg\psi$, $\psi \wedge \varphi$ are formulas in $\mathcal{P}(\mathcal{N})$. We also use $\psi \vee \varphi$ as an abbreviation for $\neg(\neg\psi \wedge \neg\varphi)$, $\psi \Rightarrow \varphi$ as an abbreviation for $\neg\psi \vee \varphi$, and $\psi \Leftrightarrow \varphi$ as an abbreviation for $\psi \Rightarrow \varphi \wedge \varphi \Rightarrow \psi$.*

In order to define a formal semantics of $\mathcal{P}(\mathcal{N})$ formulas, we consider interpretation functions $\mathcal{I}$ that map propositional atoms into {true, false}, feature names into values in their domain, and assign propositional values to numerical constraints and composite formulas according to the intended semantics.

DEFINITION 3 (INTERPRETATION AND MODELS). *An interpretation $\mathcal{I}$ for $\mathcal{P}(\mathcal{N})$ is a function (denoted as a superscript $\cdot^{\mathcal{I}}$ on its argument) that maps each atom in $\mathcal{A}$ into a truth value $A^{\mathcal{I}} \in$*

*An extended version of this paper is available as Technical Report at http://sisinflab.poliba.it/PDF/when_price_is_not_enough.pdf

$\{$*true*, *false*$\}$, *each feature name* f *into a value* $f^{\mathcal{I}} \in D_f$, *and assigns truth values to formulas as follows:*
• $(fck)^{\mathcal{I}} =$ *true iff* $f^{\mathcal{I}}ck$ *is true in* D_f, $(fck)^{\mathcal{I}} =$ *false otherwise*
• $(\neg\psi)^{\mathcal{I}} =$ *true iff* $\psi^{\mathcal{I}} =$ *false*, $(\psi \wedge \varphi)^{\mathcal{I}} =$ *true iff both* $\psi^{\mathcal{I}} =$ *true and* $\varphi^{\mathcal{I}} =$ *true, etc., according to truth tables for propositional connectives.*
Given a formula φ *in* $\mathcal{P}(\mathcal{N})$, *we denote with* $\mathcal{I} \models \varphi$ *the fact that I assigns true to* φ. *If* $\mathcal{I} \models \varphi$ *we say* $\mathcal{I}$ *is a* model *for* φ, *and I is a model for a set of formulas when it is a model for each formula.*

Clearly, an interpretation $\mathcal{I}$ is completely defined by the values it assigns to propositional atoms and numerical features. Given a set of formulas $\mathcal{T}$ in $\mathcal{P}(\mathcal{N})$ (representing an ontology), we denote *model* for $\mathcal{T}$ as $\mathcal{I} \models \mathcal{T}$. An ontology is *satisfiable* if it has a model. $\mathcal{T}$ logically implies a formula φ, denoted by $\mathcal{T} \models \varphi$ iff φ is true in all models of $\mathcal{T}$. We denote with $\mathcal{M}_{\mathcal{T}} = \{\mathcal{I}_1, \ldots, \mathcal{I}_n\}$, the set of all models for $\mathcal{T}$, and omit the subscript when no confusion arises.

The following remarks are in order for the concrete domains of our e-marketplace-oriented scenarios:
1. domains are *discrete*, with a *uniform* discretization step ϵ.
2. domains are *finite*; we denote with $\max\{D_f\}$ and $\min\{D_f\}$ the maximum and minimum values of each domain D_f.
3. even for the same feature name, concrete domains are *marketplace dependent*. Let us consider `price` in two different marketplace scenarios: pizzas and cars. For the former one, the discretization step ϵ is the €-cent, for the latter it can be fixed to *e.g.*, 100 €.

DEFINITION 4 (SUCCESSOR/PREDECESSOR). *Given two contiguous elements* k_i *and* k_{i+1} *in a concrete domain* D *we denote:*
• $s(\cdot)$ *the successor function:* $s(k_i) = k_{i+1} = k_i + \epsilon$
• $p(\cdot)$ *the predecessor function:* $p(k_{i+1}) = k_i = k_{i+1} - \epsilon$

3. MULTI ISSUE BILATERAL NEGOTIATION IN $\mathcal{P}(\mathcal{N})$

Following [5], we use logic formulas in $\mathcal{P}(\mathcal{N})$ to model the buyer's demand and the seller's supply. Relations among issues, both propositional and numerical, are represented by a set $\mathcal{T}$ – for Theory – of $\mathcal{P}(\mathcal{N})$ formulas. In a typical bilateral negotiation scenario, the issues within both the buyer's request and the seller's offer can be split into *strict requirements* and *preferences*. In our framework we call strict requirements *demand/supply*.

DEFINITION 5 (DEMAND, SUPPLY, AGREEMENT). *Given an ontology* $\mathcal{T}$ *represented as a set of formulas in* $\mathcal{P}(\mathcal{N})$ *representing the knowledge on a marketplace domain*
- *a buyer's* demand *is a formula* β *(for Buyer) in* $\mathcal{P}(\mathcal{N})$ *such that* $\mathcal{T} \cup \{\beta\}$ *is satisfiable.*
- *a seller's* supply *is a formula* σ *(for Seller) in* $\mathcal{P}(\mathcal{N})$ *such that* $\mathcal{T} \cup \{\sigma\}$ *is satisfiable.*
- $\mathcal{I}$ *is a* possible deal *between* β *and* σ *iff* $\mathcal{I} \models \mathcal{T} \cup \{\sigma, \beta\}$, *that is,* $\mathcal{I}$ *is a model for* $\mathcal{T}$, σ, *and* β. *We also call* $\mathcal{I}$ *an* agreement.

If seller and buyer have set strict attributes that are in conflict with each other, that is $\mathcal{M}_{\mathcal{T}\cup\{\sigma,\beta\}} = \emptyset$, the negotiation ends immediately because, it is impossible to reach an agreement. Now we define the utility function based on agents' preferences.

DEFINITION 6 (PREFERENCES). *The buyer's* negotiation preferences $\mathcal{B} \doteq \{\beta_1, \ldots, \beta_k\}$ *are a set of formulas in* $\mathcal{P}(\mathcal{N})$, *each of them representing the subject of a buyer's preference, and a utility function* $u_\beta : \mathcal{B} \rightarrow \Re^+$ *assigning a utility to each formula, such that* $\sum_i u_\beta(\beta_i) = 1$.
Analogously, the seller's negotiation preferences $\mathcal{S} \doteq \{\sigma_1, \ldots, \sigma_h\}$ *are a set of formulas in* $\mathcal{P}(\mathcal{N})$, *each of them representing the subject of a buyer's preference, and a utility function* $u_\sigma : \mathcal{S} \rightarrow \Re^+$ *assigning a utility to each formula, such that* $\sum_j u_\sigma(\sigma_j) = 1$.

As usual, both agents' utilities are normalized to 1 to eliminate outliers, and make them comparable. Since we assumed that utilities are additive, the *preference utility* is just a sum of the utilities of preferences satisfied in the agreement.

DEFINITION 7 (PREFERENCE UTILITIES). *Let* $\mathcal{B}$ *and* $\mathcal{S}$ *be respectively the buyer's and seller's preferences, and* $\mathcal{M}_{\mathcal{T}\cup\{\alpha,\beta\}}$ *be their agreements set. The* preference utility *of an agreement* $\mathcal{I} \in \mathcal{M}_{\mathcal{T}\cup\{\alpha,\beta\}}$ *for a buyer is defined as:*

$$u_{\beta,\mathcal{P}(\mathcal{N})}(\mathcal{I}) \quad \doteq \quad \Sigma\{u_\beta(\beta_i) \mid \mathcal{I} \models \beta_i\}$$

where $\Sigma\{\ldots\}$ *stands for the sum of all elements in the set. In an analogous way* $u_{\sigma,\mathcal{P}(\mathcal{N})}(\mathcal{I})$ *is defined for* σ.

Note that while considering numerical features, it is still possible to express strict requirements as ***reservation value*** [6]. Both buyer and seller have their own reservation values on each feature involved in the negotiation process. Referring to price and the two corresponding reservation values $r_{\beta,\mathtt{price}}$ and $r_{\sigma,\mathtt{price}}$ for the buyer and the seller respectively, if the buyer expresses $\mathtt{price} \leq r_{\beta,\mathtt{price}}$ and the seller $\mathtt{price} \geq r_{\sigma,\mathtt{price}}$, in case $r_{\sigma,\mathtt{price}} \leq r_{\beta,\mathtt{price}}$ we have $[r_{\sigma,\mathtt{price}}, r_{\beta,\mathtt{price}}]$ as a **Z**one **O**f **P**ossible **A**greement — $ZOPA(\mathtt{price})$, otherwise no agreement is possible [6]. More formally, given an agreement $\mathcal{I}$ and a feature f, $f^{\mathcal{I}} \in ZOPA(f)$ must hold. Obviously, the reservation value is considered as private information and will not be revealed to the other party, but will be taken into account by the mediator when the agreement will be computed. In order to formally define a Multi-issue Bilateral Negotiation problem in $\mathcal{P}(\mathcal{N})$, the only other elements we still need to introduce are the *disagreement thresholds*, also called disagreement payoffs, t_β, t_σ. They are the minimum utility that each agent requires to pursue a deal.

DEFINITION 8 (MBN-$\mathcal{P}(\mathcal{N})$). *Given a* $\mathcal{P}(\mathcal{N})$ *set of axioms* $\mathcal{T}$, *a demand* β *and a set of buyer's preferences* $\mathcal{B}$ *with utility function* $u_{\beta,\mathcal{P}(\mathcal{N})}$ *and a disagreement threshold* t_β, *a supply* σ *and a set of seller's preferences* $\mathcal{S}$ *with utility function* $u_{\sigma,\mathcal{P}(\mathcal{N})}$ *and a disagreement threshold* t_σ, *a* ***M****ulti-issue* ***B****ilateral* ***N****egotiation problem (MBN) is finding a model* $\mathcal{I}$ *(agreement) such that all the following conditions hold: (a)* $\mathcal{I} \models \mathcal{T} \cup \{\sigma, \beta\}$; *(b)* $u_{\beta,\mathcal{P}(\mathcal{N})}(\mathcal{I}) \geq t_\beta$; *(c)* $u_{\sigma,\mathcal{P}(\mathcal{N})}(\mathcal{I}) \geq t_\sigma$.

Observe that not every agreement $\mathcal{I}$ is a solution of an MBN, if either $u_\sigma(\mathcal{I}) < t_\sigma$ or $u_\beta(\mathcal{I}) < t_\beta$. Also notice that, since reservation values on numerical features are modeled in β and σ as strict requirements, for each feature f, the condition $f^{\mathcal{I}} \in ZOPA(f)$ always holds by condition (a).

4. UTILITIES FOR NUMERICAL FEATURES

For each feature two utility functions are needed; one for the buyer — $u_{\beta,f}$, the other for the seller — $u_{\sigma,f}$, these functions have to satisfy at least the basic properties enumerated below. For the sake of conciseness, we write u_f when the same property holds both for $u_{\beta,f}$ and $u_{\sigma,f}$. :
• Since u_f is a utility function, it is normalized to $[0\ldots,1]$. Given the pair $\langle f, D_f\rangle$, it must be defined over the domain D_f.
• We note the buyer is happier as the price decreases whilst the seller is sadder. Hence, u_f has to be monotonic and whenever $u_{\beta,f}$ increases then $u_{\sigma,f}$ decreases and vice versa.
• There is no utility for the buyer (seller) if the agreed value on price is greater (less) or equal than its reservation value. Since concrete domains are finite, for the buyer the best possible price is $\min\{D_{\mathtt{price}}\}$ whilst for the seller is $\max\{D_{\mathtt{price}}\}$.

DEFINITION 9 (FEATURE UTILITIES). *Let* $\langle f, D_f\rangle$ *be a pair made of a feature name* f *and a concrete domain* D_f *and* r_f *be a reservation value for* f. *A* **feature utility function** $u_f : D_f \rightarrow [0\ldots,1]$ *is a monotonic function such that*
– *if* u_f *monotonically increases then*

$$\begin{cases} u_f(v) = 0, v \in [\min\{D_f\}, r_f] \\ u_f(\max\{D_f\}) = 1 \end{cases}$$

– *if u_f monotonically decreases then*

$$\begin{cases} u_f(v) = 0, v \in [r_f, \max\{D_f\}] \\ u_f(\min\{D_f\}) = 1 \end{cases}$$

Given a buyer and a seller, if $u_{\beta,f}$ increases then $u_{\sigma,f}$ decreases and vice versa.

5. COMPUTING A PARETO AGREEMENT IN $\mathcal{P}(\mathcal{N})$

Obviously among all possible agreements that we can compute given a theory $\mathcal{T}$, we are interested in agreements that are Pareto-efficient. We now outline how an actual solution can be found solving a multi objective optimization problem.

First of all, let $\{B_1,\ldots,B_k,S_1,\ldots,S_h\}$ be $k+h$ new propositional atoms, and let $\mathcal{T}' = \mathcal{T} \cup \{B_i \Leftrightarrow \beta_i | i = 1,\ldots,k\} \cup \{S_j \Leftrightarrow \sigma_j | j = 1,\ldots,h\}$ – that is, every preference in $\mathcal{B} \cup \mathcal{S}$ is equivalent to a new atom in $\mathcal{T}'$.

Here we define functions to be maximized to find a solution to a multi objective optimization problem. In order to formulate functions to be maximized involving preferences expressed as formulas in $\mathcal{P}(\mathcal{N})$, let $\{b_1,\ldots,b_k\}$ the (0,1)-variables one-one with $\{B_1,\ldots,B_k\}$ and similarly $\{s_1,\ldots,s_h\}$ for $\{S_1,\ldots,S_h\}$. The function representing buyer's utility over preferences can hence be defined as:

$$u_{\beta,\mathcal{P}(\mathcal{N})} = \sum_{i=1}^{k} b_i u_\beta(\beta_i)$$

Similarly for the seller we define $u_{\sigma,\mathcal{P}(\mathcal{N})}$. As highlighted in Section 4, also utilities over numerical features have to be taken into account while finding the best solution for both the buyer and the seller. Hence, for each feature f involved in the negotiation process we have a ***feature utility function*** for the buyer $u_{\beta,f}$ and one for the seller $u_{\sigma,f}$. For instance, if we consider `price` we likely will have:

$$u_{\beta,\texttt{price}}(v) = \begin{cases} 1 - \frac{v - \max\{D_{\texttt{price}}\}}{r_{\beta,\texttt{price}} - \max\{D_{\texttt{price}}\}} \\ 0 \end{cases}$$

$$u_{\sigma,\texttt{price}}(v) = \begin{cases} 1 - \frac{v - \min\{D_{\texttt{price}}\}}{r_{\sigma,\texttt{price}} - \min\{D_{\texttt{price}}\}} \\ 0 \end{cases}$$

Given the objective functions to be optimized, in order to compute a Pareto agreement we reduce to a multi objective optimization problem (MOP). Here the functions to be optimized are utility functions both of the buyer and of the seller. Usually, in a MOP we have a set of constrained numerical variables and a set of functions to be maximized/minimized. In our setting, we have three different kind of constraints: the ontology $\mathcal{T}$, strict requirements β,σ, utility thresholds t_β,t_σ. We represent all of them as linear constraints.

Obtain a set of clauses $\mathcal{T}''$ s.t. each clause contains only one single numerical constraint and $\mathcal{T}''$ is satisfiable iff $\mathcal{T}' \cup \{\sigma, \beta\}$ does. In order to have such clauses, if after using standard transformations in clausal form [3] you find a clause with two numerical constraints $\chi : A \vee \ldots (f_i c_i k_i) \vee (f_j c_j k_j)$ pick up a new propositional atom $\overline{A}$ and replace χ with the set of two clauses.

$$\left\{ \begin{array}{l} \chi_1 : \overline{A} \vee A \vee \cdots \vee (f_i c_i k_i), \\ \chi_2 : \neg\overline{A} \vee A \vee \cdots \vee (f_j c_j k_j) \end{array} \right\}$$

As a final step, for each clause, replace $\neg(f \leq k)$ with $(f \geq s(k))$ and $\neg(f \geq k)$ with $(f \leq p(k))$. Use a modified version of well-known encoding of clauses into linear inequalities (*e.g.*, [4, p.314]) so that every solution of the inequalities identifies a model of $\mathcal{T}''$. If we identify true with values in $[1 \ldots \infty[$ and false with values in $[0 \ldots 1[$ each clause can be rewritten in a corresponding inequality.

- map each propositional atom A occurring in a clause χ with a (0,1)-variable a. If A occurs negated in χ then substitute $\neg A$ with $(1-a)$, otherwise substitute A with a.
- replace $(f \leq k)$ with $\frac{1}{\max\{D_f\}-k}(\max\{D_f\} - f)$ and $(f \geq k)$ with $\frac{1}{k} f$.

6. THE BARGAINING PROCESS

Summing up, the negotiation process covers the following steps:
Preliminary Phase.The buyer defines strict β and preferences $\mathcal{B}$ with corresponding utilities $u_\beta(\beta_i)$, as well as the threshold t_β, and similarly the seller σ, $\mathcal{S}$, $u_\sigma(\sigma_j)$ and t_σ. Both agents inform the mediator about these specifications and the $\mathcal{T}$ they refer to.
Negotiation-Core phase. For each $\beta_i \in \mathcal{B}$ the mediator picks up a new propositional atom B_i and adds the axiom $B_1 \Leftrightarrow \beta_i$ to $\mathcal{T}$, similarly for $\mathcal{S}$. Then, it transforms all the constraints modeled in β, σ and (just extended) $\mathcal{T}$ in the corresponding linear inequalities following the procedures illustrated in Section 5. Given the preference utility functions $u_{\beta,\mathcal{P}(\mathcal{N})}$ and $u_{\sigma,\mathcal{P}(\mathcal{N})}$, the mediator adds to this set of constraints the ones involving disagreement thresholds $u_{\beta,\mathcal{P}(\mathcal{N})} \geq t_\beta$ and $u_{\sigma,\mathcal{P}(\mathcal{N})} \geq t_\sigma$. With respect to the above set of constraints, the mediator solves a MOP maximizing the preference utility functions $u_{\beta,\mathcal{P}(\mathcal{N})}$, $u_{\sigma,\mathcal{P}(\mathcal{N})}$ and for each feature f involved in the negotiation process also the feature utility functions $u_{\beta,f}$ and $u_{\sigma,f}$. The solution returned as solution to the MOP is the agreement proposed to the buyer and the seller. Notice that a solution to a MOP is always Pareto optimal [2]. The participants can then either accept or reject the proposed agreement, in a classical one-shot fashion.

Let us present a tiny example in order to better clarify the approach. Given the toy ontology in $\mathcal{P}(\mathcal{N})$,

$$\mathcal{T} = \begin{cases} \texttt{ExternalColorBlack} \Rightarrow \neg\texttt{ExternalColorGray} \\ \texttt{SatelliteAlarm} \Rightarrow \texttt{AlarmSystem} \\ \texttt{NavigatorPack} \Leftrightarrow \texttt{SatelliteAlarm} \wedge \texttt{GPS\_system} \end{cases}$$

the buyer and the seller specify their strict requirements and preferences:

β = Sedan $\wedge$ (price $\leq$ 30,000) $\wedge$ (km_warr $\geq$ 120,000) $\wedge$ (year_warr $\geq$ 4)
β_1= GPS_system $\wedge$ AlarmSystem
β_2= ExternalColorBlack $\Rightarrow$ Leather_seats
β_3= (km_warr $\geq$ 140,000)
$u_\beta(\beta_1)$ = 0.5 , $u_\beta(\beta_2)$= 0.2 , $u_\beta(\beta_3)$= 0.3 , t_β=0.2

σ = Sedan $\wedge$ (price $\geq$ 20,000) $\wedge$ (km_warr $\leq$ 160,000) $\wedge$ (year_warr $\leq$ 6)
σ_1= GPS_system $\Rightarrow$ (price $\geq$ 28,000)
σ_2= (km_warr $\leq$ 150,000) $\vee$ (year_warr $\leq$ 5)
σ_3= ExternalColorGray
σ_4= NavigatorPack
$u_\sigma(\sigma_1)$]= 0.2, $u_\sigma(\sigma_2)$= 0.4, $u_\sigma(\sigma_3)$= 0.2, $u_\sigma(\sigma_4)$= 0.2, t_σ=0.2

Then the final agreement is:

$$\begin{aligned} \mathcal{I} \;\; : \;\; & \{\texttt{Sedan}^{\mathcal{I}} = \text{true}, \texttt{ExternalColorGray}^{\mathcal{I}} = \text{true}, \\ & \texttt{SatelliteAlarm}^{\mathcal{I}} = \text{true}, \texttt{GPS\_system}^{\mathcal{I}} = \text{true}, \\ & \texttt{NavigatorPack}^{\mathcal{I}} = \text{true}, \texttt{AlarmSystem}^{\mathcal{I}} = \text{true}, \\ & \texttt{price}^{\mathcal{I}} = 28{,}000, k^{\mathcal{I}} = 160{,}000, \texttt{year\_warr}^{\mathcal{I}} = 5\} \end{aligned}$$

Here, for the sake of conciseness, we omit propositional atoms interpreted as false.

7. REFERENCES

[1] F. Baader and P. Hanschke. A schema for integrating concrete domains into concept languages. In *proc. of IJCAI-91*, 1991.
[2] F. Hillier and G. Lieberman. *Introduction to Operations Research*. 2005.
[3] D. W. Loveland. *Automated theorem proving: A logical basis*. North-Holland, 1978.
[4] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.
[5] A. Ragone, T. Di Noia, E. Di Sciascio, and F. Donini. A logic-based framework to compute pareto agreements in one-shot bilateral negotiation. In *Proc. of ECAI'06*, 2006.
[6] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, 1982.