

多様なインセンティブを含む売買制約ルールの表現と動的制約代数による検索支援

Applications of Dynamic Constraint Algebra to E-Commerce with Various Incentives

小澤正幸[†] 岩井原 瑞穂^{††} 上林 彌彦^{††}

MASAYUKI KOZAWA,[†] MIZUHO IWAIHARA^{††}
and YAHIKO KAMBAYASHI^{††}

商取引の売買条件には、売り手側が商品を売り切るために多様なインセンティブ（報償金）を設定することがあり、買い手側は商品内容と価格だけではなく、商品ごとに異なるインセンティブを比較する必要がある。多様なインセンティブを含む売買制約は従来の関係データベースでは扱いが困難である。本稿ではこのような複雑なインセンティブが用いられる例として航空券を取り上げ、動的制約データベースと呼ばれる関係データベースの拡張モデルを応用することでこの問題に取り組む。まず、実際の航空券販売サイトから得られたデータを元に、このモデルで扱える制約条件・ルールについて考察する。そして、このルールを動的制約データベースエンジンで処理し、購入する航空券の選択を支援するシステムの実装を行う。

1. はじめに

電子商取引では単に価格のみによらずに商品の価値が決まることがある。そういった場合、現状では自然言語で記述されているそれら複雑な売買条件やルールを考慮し、それを扱えるようにする必要がある。動的制約データベース (DCDB: Dynamic Constraint Database) はこのような問題を扱うためのモデルであり、動的制約代数 (DCA: Dynamic Constraint Algebra) はその質問言語である⁵⁾⁶⁾。

商品をすべて売り切り、全体での利益率を向上させるために、売り手が様々な優遇条件を設定することが行なわれている。たとえば、買い手の購入した品物が変更やキャンセルされることにより売れ残りが生じるのを防ぐために、変更やキャンセルを禁止したり高い手数料を設定する見返りに価格を低くする、などのインセンティブ（報償金）を設定し、買い手を誘導するなどの例がある。同一内容の商品でも、多種のインセンティブが設定されると、その組み合わせにより複雑な

価格構成が生じうる。インセンティブは売り手側の都合であり、買い手側の都合とは必ずしも一致しない。買い手側は、商品内容の他に、これらのインセンティブを組合せることで有利な価格を得ることになる。またインセンティブの設定方法は、一般に売り手ごとに異なるため、買い手側は慎重にインセンティブを比べなければならない。

DCDB は、このような“商品ごとに”異なるような条件を記述することに適した性質を持っている。従来の RDB による電子商取引サイトでは、複雑な条件はプログラムとして表現され、あらかじめ規定した方法でのみしか条件の指定ができなかった。DCDB ではこのような外部プログラムで表現されていたような複雑なロジックをプログラムから分離し、タプルを持つデータの一部（制約データベースの術語では“制約”）として表現することができる。そしてこれにより、多様なインセンティブ、つまり複雑な売買条件やルールを、商品ごとに柔軟に設定することができ、また質問言語 DCA を用いて、制約による柔軟な検索や演算を行うことが可能となる。

まず理論的背景として DCDB について概説し、次にインセンティブについて述べる。続いて実際に電子商取引が行われている物の中で、もっともインセンティブが多用されている実例として航空券販売を取り上げ、その具体的な性質について述べ、それをどのよ

[†] 京都大学工学部

Faculty of Engineering, Kyoto University
kozawa@db.soc.i.kyoto-u.ac.jp

^{††} 京都大学情報学研究科社会情報学専攻

Department of Social Informatics Graduate School of Informatics, Kyoto University
{iwaihara, yahiko}@db.soc.i.kyoto-u.ac.jp

うに制約条件で表現したかを説明する。そして最後に構成されたデータに対し、有用な質問の例を示す。

2. 動的制約データベース

制約データベース⁷⁾は関係のタプルごとに制約式と呼ばれる論理式を持つ。この制約式を充足する値の組み合わせが、通常の関係のタプルに相当する。ただし制約データベースでは制約式を満足する値の組み合わせが無限に存在する場合も有限のサイズで表現できるなど、通常の関係よりも高い表現力を持つ。

関係のスキーマは関係データベースと同様に $[A_1, \dots, A_k]$ で与えられるが、そのタプルは $[a_1, \dots, a_k]/\phi$ で与えられる。 a_i が属性 A_i の値であり、これは定数或いは制約変数である。また、 ϕ が制約式である。制約式としては、充足判定問題が可解である様々な制約式を用いることが出来る。本稿では等号論理、つまり“制約変数同士”、あるいは“制約変数と定数”を等号かその否定 ($=, \neq$) で結んだ項 (e.g. $x = 1, x \neq y, d = \text{“USD 120”}$) をさらに論理演算子 \wedge (and), \vee (or), \neg (not), \rightarrow (implication) で結合した制約式を用いる。制約関係の例を表 1 に示す。

2.1 動的制約代数

DCA は、これまでに知られている制約関係代数について変数によるスキーマ定義を行わず、任意の変数が混在する制約式も取り扱えるようにしたものであり、これにより、多様な電子商取引の制約条件を柔軟に扱うことを可能にしている。また、あらかじめ出現する変数を限定していないために、変数を陽に指定せずに限量子を与える演算として動的制約演算を新たに導入している。以下に DCA の各演算とその大まかな説明を示す。

集合演算 通常の場合同士の演算に対し制約を扱えるように拡張した物である。いずれもリレーションに対する二項演算である。

和集合演算 (\cup) 二つのリレーションに対し和集合を取る。

差集合演算 ($-$) 差集合を取る。

直積集合演算 (\times) 直積集合を取る。このとき、二つのタプルの制約式の論理積が新しいタプルの制約式となる。

関係演算 関係代数の演算に対し、各タプルの制約式の論理演算を行うよう拡張がされている。代入演算に関しては制約変数が存在する制約データベース独特の物である。いずれもリレーションに対する単項演算である。

選択演算 (σ_ρ) 条件 ρ と各タプルの制約式の論

理積を取ったリレーションを得る。

射影演算 (π_S) 属性の部分集合 S のみからなるスキーマのリレーションを得る。

代入演算 (S_θ) 変数集合から定数集合への部分写像 θ を用いて制約変数への具体的な定数値の代入を行う。

動的制約演算 DCA 特有の演算である。限量子を用いた演算の代用である。以下の三種類があり、いずれもリレーションに対する単項演算である。

Boolean 演算 (B_T, B_M) 制約の充足性を判定する演算である。前者は制約条件が恒真であるタプルを、後者は真偽の定まっていない、つまり充足可能かつ非充足可能であるタプルからなるリレーションを得る。(恒偽の、つまり制約を充足するような値の組み合わせが存在しないようなタプルは、制約データベースではリレーションから除かれる。)

Cardinality 演算 ($C_{|A|\alpha k}$) 制約の充足解の個数に対する演算である。属性 A が持ちうる値の個数 $|A|$ が、等・不等号 $\alpha (\in \{<, =, >\})$ と、非負整数あるいは無限大 ∞ を表す記号 “inf” である k により設定される条件を設定し、それを満たすタプルのみからなる集合を得る。

Dependency 演算 ($D_{cv(A)\beta Dep}, D_{x\beta Dep}$) 制約への依存性を調べる演算である。それぞれ、前者の形式では属性 A の値が、後者の形式では変数 x が、そのタプルに付された制約式に対して、 β が \in ならば依存するタプルのみを、 \notin ならば依存しないタプルのみを得る。なお、“制約式 ϕ が変数 x に依存する”、とは x のみに異なる値を割り当て、しかも ϕ に代入を施した結果の真偽値が互いに異なるような二種の代入写像 θ_1, θ_2 が存在することで定義される。

3. インセンティブ付商品の電子商取引

文献⁵⁾⁶⁾では、商品の組み合わせの制限やそれらによる割引などの商条件をを制約で記述して、検索及び照合を行うための基礎的な枠組みとして DCDB が提案されている。本論文では対象として、商品を単品で購入する場合を扱っているが、単純なカタログ販売型商取引のように価格が固定されているのではなく、各種のインセンティブによりこれら変動する場合の表現モデルを考察する。つまりカタログ販売型の形態でも非価格的条件を計算可能な形式で表現することが

pid	Item	Model	Price	Buy	Condition
p0	MPU	m_1	p_1	b_1	$(m_1 = 500 \vee m_1 = 750) \wedge (m_1 = 500 \rightarrow p_1 = 249) \wedge (m_1 = 750 \rightarrow p_1 = 399)$
p0	RDRAM	m_2	p_2	b_2	$m_2 = 128\text{MB} \vee m_2 = 256\text{MB}$
p0	SDRAM	128MB	100	b_3	T
p0	Fan	Twin	60	b_4	$m_1 = 750$
p0	Fan	Single	p_3	b_5	$(b_1 = 1 \rightarrow p_3 = 30) \wedge (b_1 = 0 \rightarrow p_3 = 40)$

表 1 制約関係

目的である。

3.1 インセンティブ

商取引においては様々な売買ルールやインセンティブが設定されている。まず、これらの分類を行う。インセンティブルールは売買ルールの一部であり、これは売買ルールのうちで買い手を誘導する性質があるものと定義できる。つまり比較することにより購入する商品の選択に影響を与えうるものと言い換えられる。ルールの内容に対して以下のような分類を行うことができる。

選択条件ルール (selection rule) 商品に関して買い手が選ぶことのできる選択肢を表わすルールであり、可能な選択肢および選択肢ごとの価格、あるいは商品の組み合わせに対する条件を表現する。航空券では、利用可能な曜日、期間、便名などが相当する。選択肢に応じて異なる価格が設定されることがよく行われる。選択肢には、平日のみ有効など、限定された条件を与える代わりに、低い価格を提供するものがある。

違約金ルール (penalty rule) 商品の契約内容を買手が変更する場合に適用される違約金に関するルールである。選択条件ルールの一種と考えられる。ホテルやレンタカー、航空便のキャンセル手数料や予約変更手数料がこれに相当する。価格は低いが違約金は高い商品や、逆に価格は高いが違約金は低い商品が存在し、変更に対する柔軟性という付加価値の観点から違約金ルールについても比較する必要がある。

資格的ルール (qualification rule) ある商品の購入において、買い手が満たすべき資格的な条件である。幼児割引や、酒類は 20 才以上などがこれに該当する。買い手が購入できる商品であるかどうかの条件であり、買い手が資格を満たしているかを売り手が確認するか、買い手が自己申告を行う。法令により、各社共通の資格を定めている場合もあれば、会社により異なる資格を定めている場合もある。後者の例として、レンタカーの

同乗者による運転を配偶者あるいは同居人ならば無料にするなど、会社ごとに異なるため、一般には資格的ルールも比較する必要がある。

例外ルール (exception rule) 通常の売買ルールの体系とは別に限られた範囲・期間にのみ適用される例外的な条件である。懸賞やクーポン、プロモーションコードによる割引、特売、期間限定の手数料サービス、特定の商品の割引など、売り手側の販売促進策からの特別割引などであり、買い手側から見た場合、他の条件に制限が加わらないように見えることが多い。通常の価格が固定されたカタログ販売型の電子商取引でも頻繁に出現する。

これらルールの性質からの分類とは別に、各々の条件に関して、そのルールの論理構造や値の取り方からも分類を行うことができる。この分類はルールの計算を行う上で重要となる。

性質からの分類としては、各々、とりうる値の候補を示すもの、条件同士の組み合わせを指定するもの、「A ならば B」のように含意を示すものの三通りとその組み合わせがある。それぞれ順に OR、AND、IMPLY を用いて表現できる。

また、条件の内容から分類を行うと、条件の強弱あるいは厳しさと緩さに関して、全順序付き集合に写像できる物、つまり強弱を実数値など扱いやすい他の全順序付き集合に対応させることで大小比較できる物と、とりうる値の候補などが有限集合で表され、制限の強弱がこの集合の部分集合への絞り込みで表現される物とがある。前者を“全順序型”の条件、後者を“半順序型”の条件と呼ぶことにする。

全順序型の条件の場合、条件の比較はこの順序関係を元に行う。また条件の指定方法も、具体的な値とそれに対して「等しい」「以上」「以下」などの比較演算を指定することで行われる。しかし、半順序型の条件では単純に要素数や包含関係などによる集合の大小で比較を行うよりも、買い手側の希望する要素がその集合に含まれるかどうかなどを基準に、条件の厳しさと適合性を見るべきである。

3.2 航空券の売買制約

現在、実際に電子商取引で扱われているものの中で、通常の価格構成においてもインセンティブが多用されている顕著な例が航空券販売である。その典型的な例は早期購入割引や閑散割引などである。その他にも様々なルールがあり、これらの制限の強弱や有無が価格に影響を与え、選択条件ルールや違約金ルールなどとして働いている。またその一方、航空会社の担当者の判断で割引が行なわれることにより、座席数が違うだけで、買い手の見た目には価格しか変わらず、それ以外は全く同じ条件で売られている券、つまり例外ルールとしてのインセンティブが作用しているチケットもある。結局のところ価格だけを基準に券を購入することはできず、複雑なルールを読み比べる必要がある。

航空券販売サイトでは以下のような流れで予約を行うのが一般的である。

- (1) 区間と日付の指定を行う。
- (2) 料金順でソートされたチケットが一覧表示される。チケットの違い(クラス¹の違い)は主に航空会社と付されたインセンティブにより生じる。
- (3) 利用者は自然言語で記述されたインセンティブを読み比べて購入するチケットを選択する。

(2)においては区間により、何十種という選択肢が提示され、付されているインセンティブがどう異なるのかを見比べることは容易ではない。このインセンティブを制約式に置き換えることで論理的な処理を可能とし、買い手がこの選択肢の中から最適なチケットを選択するのを支援できるようにする。

3.3 チケットルールの例

航空券の予約販売サイトである Travelocity.com² からロサンゼルス・ニューヨーク間往復航空券のデータを収集し、DCDB用のデータに変換した。前節で述べたとおり、航空券には返金・変更の可否をはじめ有効期限などで複雑で様々な代償型インセンティブがあり、これらの違いによって様々な料金でチケットが売られている。たとえば今回データを集めたロサンゼルス・ニューヨーク間では80種近い往復チケットが存在した。

航空券に付けられているルールは「チケットが有効である、あるいは有効になる時期について記述するも

の」「購入や変更・キャンセルなどの手続きに関するもの」「割り増し・割引などに関する条件」「他の路線と組み合わせて購入するときの条件」「その他」に大別することができた。

インセンティブとして作用し、チケット選択の判断材料となり价格的な差を生み出す要因となっていたのはこれらのうち「時期的な条件」と「手続きに関する条件」に関するものである。³

3.4 制約の表現

各種の条件を制約式として表現するに際して、資格的ルールの大半など同じ航空会社であれば一定であるルール、つまり会社間では差が存在するが、同一会社のチケットに対して局所的に見た場合に固定的である条件が多いことから、同じ会社で価格の異なるチケットを一つのタブルにまとめあげ、異なる部分を制約式で記述した。このようにまとめることで商品の内容が同一であるものに対しルールの違いのみが価格に影響を与えている様子が分かりやすくなる。

なお、小児割引に関する条件などがそうであったが、完全に航空券販売において共通であったルールなどはそれに対する要求が「適切に表示されること」のみであるため今回のデータ処理では割愛した。

- (1) 全チケットに渡って存在する基本的な情報を属性とし、それ以外の個別の条件などは制約式の内部で表現した。⁴
- (2) 同じ航空会社のチケット間で、違約金ルールや資格的ルールなど固定的である項目は定数で、そうではなく選択条件ルールや例外ルールなど、インセンティブにより変動する値は変数とそれがとりうる値で表現した。
- (3) 候補値の提示、条件の組み合わせ、含意の表現に関しては3.1で述べたとおりそれぞれOR, AND, IMPLYで表現した。
- (4) ある属性に対応するルールが、元のデータに記載されていない場合、それが“制限がない”ことを表している、すなわち任意の値で構わない場合は制約のないダミー変数、つまり制約式中に現れない変数を値とし、値が明示的に存在しない場合と区別した。
- (5) チケットのクラスと価格が一对一对応であることから価格をキーのように扱った。

¹ 航空券販売では座席を同じ便の同じ等級であってもそれを細分し、それらに異なる条件を付けていわば別の商品として売ることが行われている。この細分化された一つ一つを“クラス”と呼ぶ慣習がある。

² <http://www.travelocity.com/>

³ 「他の路線と組み合わせて購入するときの条件」は処理するべきデータ量の問題から将来の研究課題とした。

⁴ DCDBでは制約データベースにおける“スキーマ定義された制約変数”が属性に、DCDB特有の“スキーマ定義されず動的に現れた変数”がそれ以外の変数に対応する。

構成した制約関係のスキーマを表 2 に、テーブルの一部分を表 3 に示す。実際に作成されたテーブルと比較し、紙面の都合によりダブル数のみならず属性や制約式に対しても多少の省略がなされている。また、表 3 では制約変数は斜体で、制約式中に現れる定数は二重引用符に囲って表示した。なお、制約式中の $v = \{c_1|c_2|\dots|c_k\}$ のような波括弧を用いた表記は $(v = c_1 \vee v = c_2 \vee \dots \vee v = c_k)$ の略記であり、 $\{“01/12/23” \dots “01/12/31”\}$ のような、波括弧の中で “...” を用いた表記は、更に $\{“01/12/23”|“01/12/24”|\dots|“01/12/31”\}$ のように連続値を指定する略記である。

4. DCA での質問の例と比較の手順

ユーザーの視点からの使い方をいくつか挙げ、質問の例としていくつかを示す。

4.1 基本的な操作

基本的な検索操作と、それに対応する質問を示す。
希望に添うチケットの選択 たくさんのチケットや価格帯の中から希望に添うものを絞り込めるようにする。希望を表すためにユーザーが入力する情報として以下を想定する。

- (1) チケットに対する時間的な希望 (日付と曜日指定)
 - (2) 入金・予約・購入の条件 (今購入するか、あとで購入するかなどを指定)
 - (3) いつ帰るのかについて (すぐに帰る / 土曜の晩を越す / 30 日以上滞在する)
 - (4) 変更の可能性について (まずない / 時間は変わりうる / 行き先も変わりうる / 保留)
- 上記ををもとにそれぞれ次のような条件を参照し、求めるべきチケットを検索することを検討する。それぞれが対応する属性名を表 2 をもとに括弧内に記す。

- (1) その券が希望の日にも使えるか。(AD, AW, AT, LT)
- (2) 前もって買わなければならないチケットかどうか (AP, TI)
- (3) 復路の券が有効かどうか (MS, LR)
- (4) 変更が可能かどうか (CC, RR)

たとえば「11月21日から11月22日のどちらかは決めていないが、どちらの日でも利用できるチ

ケット」は次の選択質問を内部的に発行することにより得ることができる。

$$\sigma_{AD=“11/21” \vee AD=“11/22”}(\text{FlightList})$$

また、これらの希望を表現する制約式を構成し、その制約式だけからなる制約テーブルと売り手が提示しているカタログの制約テーブルとで結合演算 (直積と選択で表現できる。)を行うことで 2.1 で示した DCA での直積演算が制約の論理積を取るという性質から、買い手と売り手の希望をマッチングさせることができる。

こちらの方法を用いれば、選択演算を用いた場合よりも曖昧さや場合分けがある場合にも対応できる。

条件の決定 未確定であった条件、つまり複数の選択肢が残されている変数に対し、代入演算を行うことで、その条件が確定されたものであるとして処理を進めることができる。たとえば「11/21 日に出発する」事を決めれば、以下のようにすることで前節での絞り込みに加えてそれを確定したものと扱うことができる。

$$S_{AD=“11/21”}(\text{FlightList})$$

確定しているチケットを調べる $B_M(\text{FlightList})$ により「まだ決定していない条件により満たされなくなる可能性があるチケット」を、 $B_T(\text{FlightList})$ により「現時点で既に条件を満たしていることが確定しているチケット」を調べることができる。

選択肢の表示 特定の条件に対し、選択肢が残されているかどうかを知ることができる。たとえば以下のようにすることで、それぞれ「価格に選択肢があるチケット」や「最小滞在日数 (MS) が条件により変動するチケット」を求めることができる。

$$C_{1 < PR}(\text{FlightList})$$

$$D_{cv(MS) \in Dep}(\text{FlightList})$$

4.2 応用的な質問

前節の基本的な質問を組み合わせることで様々な質問を作ることができる。

応用 1 制限の有無に関してたとえば「帰る日になにか制限のあるチケット」「帰る日はオープンであるチケット」をそれぞれ次のようにして調べることができる。

$$C_{|PR| \geq 1}(S_{MS \neq “NoRestriction”}(\text{FlightList}))$$

$$C_{|PR| \geq 1}(S_{MS = “NoRestriction”}(\text{FlightList}))$$

応用 2 たとえば前売りの制限条件 (AP) と料金 (PR) に関して次のような質問を行うことで「14 日前でも 3 日前でも料金の変わらないチケット」を求めることができる。

等号ではなくその否定が用いられているときはその全体の否定を表す。
現在の動的制約代数処理系の実装では離散値のみ扱うことが可能である。

属性名	略記	意味	具体例	条件の型
Airline	AL	航空会社名		—
Price	PR	チケットの価格		全順序型
Advance Purchase	AP	前売りの制限	「出発の14日前までに購入手続きを済ませなければならない。」などの制限が存在する。	全順序型
Ticket Issue by	TI	チケットの購入期限	「2001/11/10までに購入しなければならない」などの制限がある。	全順序型
Available Date	AD	利用できる日付		半順序型
Available Day of the Week	AW	利用できる曜日	「火・水・木曜日のみ利用可能」などの制限が存在する。	半順序型
Available Time	AT	利用できる時間		半順序型
Latest Travel	LT	チケットの有効期限		全順序型
Minimum Stay	MS	最小滞在日数	「復路のチケットは最初の日曜日以降利用可能」などの制限が存在する。	全順序型
Latest Return	LR	最大滞在日数		全順序型
Reroute	RR	路線変更の可否	「行程の日時は変更可能であるが出発地と到着地は変えられない。」などの制限がある。	半順序型
Change Charge	CC	変更手数料		全順序型
Co-Terminals	CT	同一と見なされる空港のリスト		半順序型
Stop Over	SO	途中待機の可否		全順序型

表 2 航空券データの制約関係の属性

AL	PR	AP	AD	AW	AT	MS	制約式
AW	pr_0	21 days	ad_0		12:00N-1:00PM	ms_0	$pr_0 = \{ "33900" "38400" "45300" \}$ $\wedge (ad_0 \neq \{ "01/11/22" "01/11/23" "01/12/23" \dots "01/12/31" \})$ $\rightarrow ms_0 = "1st Sun"$ $\wedge (ad_0 = \{ "01/11/22" "01/11/23" "01/12/23" \dots "01/12/31" \})$ $\rightarrow ms_0 = "No Restriction"$
AA	pr_1	ap_1	ad_1	aw_1	at_1	1st Sun	$pr_1 = \{ "33900" "45300" "53600" \}$ $\wedge pr_1 = \{ "33900" "45300" \}$ $\rightarrow (ap_1 = "21 days" \wedge at_1 = "6:30AM-12:30AM")$ $\wedge pr_1 = "33900" \rightarrow aw_1 = \{ "Tue" "Wed" \}$ $\wedge pr_1 = "53600"$ $\rightarrow (ap_1 = "14 days"$ $\wedge aw_1 = \{ "Mon" "Tue" "Wed" "Thu" \}$ $\wedge \neg ad_1 = \{ "01/11/21" "01/11/25" "01/11/26" \})$
ATA	pr_2	ap_2	ad_2	aw_2		1 day	$pr_2 = \{ "22500" "27100" "31600" "34000" "38400" "40700" "49800" \}$ $\wedge pr_2 = \{ "22500" "31600" \} \rightarrow aw_2 = \{ "Tue" "Wed" "Sat" \}$ $\wedge pr_2 = "34000" \rightarrow ap_2 = "3 days"$

表 3 航空券データの制約関係 FlightList

$$F_{14} = S_{AP="14days"}(\text{FlightList})$$

$$F_3 = S_{AP="3days"}(\text{FlightList})$$

$$F = \sigma_{PR=PR\#2}(\sigma_{AL=AL\#2}(F_{14} \times F_3))$$

応用 3 属性間の関連・依存を調べることもできる。

ここで、「属性 A に属性 B が依存する」とは、A の異なる値に対して B にも異なる値が対応し、制約を充足すると定義する。

具体例としては「曜日 (AD) の選び方により、値

段 (PR) の変わるチケット」(すなわち、属性 AD に属性 PR が依存するチケット)のみを求めるといったことが考えられ、これは以下のような質問により検索できる。

$$C_{|AD|>1}(C_{|PR|>1}(\text{FlightList}))$$

なぜこれで依存する物を求められるかを「制約テーブル F の中から属性 A に属性 B が依存するタブルのみを得る質問」に対応する次の式を用いて説明する。

$$C_{|A|>1}(C_{|B|>1}(F))$$

これにより求まるタブルに対して、 A, B それぞれに対し制約を充足する値の組 a_1, a_2, \dots, a_m と b_1, b_2, \dots, b_n が存在し、しかも $m > 1$ と $n > 1$ が成立する。ここで、ある a_i と b_k の組に対して制約式が真になる。

ある B の値 b_k に対して、 b_k と共に充足解の一部となる A の値の集合を $A[b_k]$ とおく。 $n > 1$ であるため、 b_k 以外にも充足解の一部となる b_l が存在する。

$A[b_k]$ が a_1, \dots, a_n と一致するならばある a_i に対して a_i と b_l は充足解の一部となる。 $a_j (i \neq j)$ と b_k も充足解の一部であることから、 A は B に依存している。 $A[b_k]$ が $\{a_1, \dots, a_n\}$ に真に含まれるならば、 $\{a_1, \dots, a_n\} - A[b_k]$ の要素を一つ選び、 a'_i とすれば a'_i と $b'_l (\neq b_k)$ が充足解を構成することから、 A は B に依存する。

逆に、属性 A に属性 B が依存するならば、定義より A と B 両方が複数の値で充足できることが必要である。従ってこのクエリーにより得られるタブルにそれが含まれるのは自明である。

4.3 比較操作

現在の実装では不等号による大小比較が行えないために条件の強弱に関して比較を行う場合には工夫が必要となる。ここでは 3.1 での条件型の分類に基づき、それぞれに対してその比較手法について述べる。

全順序型条件 全順序型条件では条件と対応する実数値との対応表を持つ必要がある。データ構成の観点からはこれは条件の追加時に動的に構成されるようにするのが理想である。このテーブルがあればそれを元に「より厳しい条件」「より緩い条件」を求めることができるので、たとえば「このチケットよりも価格が安いチケットはどれか？」あるいは「前売り制限に関してこのチケットよりも条件が緩いチケットはどれか？」などを求めることができるようになる。より「厳しい」、あるいは「より緩い」条件が求めれば、それに該当する商品を

絞り込む検索には「希望に添うチケットの選択」で述べたものと同じ方法が使える。

半順序型条件 半順序型条件では条件の間に真の部分集合として含む、含まれる、という意味で条件の強弱を設定することはできる。しかし、この包含関係は半順序関係であるので必ずしも条件間に強弱を設定できるとは限らない。

たとえば、曜日制限 (AD) に関して、「どの曜日でも使える」という制限は「平日しか使えない」という制限よりも緩く、「火曜・水曜・木曜しか使えない」という制限に対してはさらに緩い。しかし、この「火曜・水曜・木曜しか使えない」に対する制限と「木曜・金曜・土曜しか使えない制限」ではどちらが厳しいかを比較することができない。

したがって、この半順序型条件に分類される場合は候補となる値の集合に対して集合間の関係の比較演算を行うことになる。

DCA の現在の実装ではとりうる値を離散値化した上でテーブルとして持つことで集合を表現し、それに対して買い手側の希望が充足するか、などを見ることになる。

5. 実装

既に実装が為されている Perl ベースの DCA 処理系に接続する形で、DCDB による航空券検索システムを Web アプリケーションとして実装を行った。

6. 関連研究

6.1 RuleML²⁾³⁾

Semantic Web の一環としてルールや論理式を XML タグでマークアップする方法を標準化するための W3C のプロジェクトである。これ自体は幅広い分野で論理式表現を扱えるようにするために XML のインターフェイスを提供するものであるが、電子商取引に関しては文献⁴⁾や IBM Common Rules などのプロジェクトからの成果を取り入れている。

6.2 Vertical Representation

文献¹⁾により提示された“垂直表現”も本稿と同様に電子商取引において様々な種類の商品を一つのデータベースシステムで扱い、商品にまつわる多種多様なデータを種類毎に属性に割り当てていく事で生じる「データ追加時の属性の増加」「空値の多いテーブル」という問題を扱ったものである。基本的なアイデアはデータをタブル単位ではなく、(タブルの ID, 属性名, 属性値) の三つ組の形式で保持するというものである。

ただしあくまでも関係データベースモデルと等価の表現であり、DCDBの方が表現能力は高い。

7. 結 論

本稿では商取引上でのインセンティブについて考察を行い、検索・比較する手法を検討した。また、その実現方法としてそれらを制約で表現し、動的制約データベースを用いた。そして複雑なインセンティブを持つ商品の代表である航空券販売に応用し、このシステムが多種の複雑な制約を取り扱う能力があることを示した。今後の研究予定としては並行して進められたユーザーインターフェイスに関する研究成果との統合など、実装の強化を中心にすすめる方針である。

参 考 文 献

- 1) Rakesh Agrawal, Amit Somani and Yirong Xu
“Storage and Quarrying of E-Commerce Data”
VLDB 2001 pp.149-158.
 - 2) Harold Boley, Said Tabet and Gerd Wagner
“Design Rationale of RuleML: A Markup Language for Semantic Web Rules” Semantic Web Working Symposium, July/ August 2001, Stanford, CA.
 - 3) Harold Boley “The Rule Markup Language: RDF-XML Data Model, XML Schema Hierarchy, and XSL transformations” INAP2001 Tokyo, Japan, in October 2001
 - 4) Benjamin N. Grosz, Yannis Labrou, Hoi Y. Chan “A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML” E-COMMERCE 99, Denver, Colorado.
 - 5) Mizuho Iwaihara “Supporting Dynamic Constraints for Commerce Negotiations” 2nd Int. Workshop in Advanced Issues of E-commerce and Web-Information Systems (WECWIS), IEEE Press, pp.12-20, June 2000.
 - 6) Mizuho Iwaihara “Matching and Deriving Dynamic Constraints for E-Commerce Negotiations” Workshop on Technologies for E-Services, (informal proceedings), Cairo, Sep. 2000.
 - 7) Gabriel Kuper, Leonid Libkin, Jan Paredaens(Eds.)
“Constraint Database” Springer-Verlog, 2000.
 - 8) Kelvin Leyton-Brown, Mark Pearson, Yoav Shoham “Towards a Universal Test Suite for Combinatorial Auction Algorithms” EC’00 October 17-20, 2000, Minneapolis, Minesota.
-