修辞構造木から自動変換した談話依存構造木の性質について

†林 克彦　　平尾 努　　吉田康久　　永田 昌明
日本電信電話株式会社 NTT コミュニケーション科学基礎研究所

†hayashi.katsuhiko@lab.ntt.co.jp

1 はじめに

修辞構造理論 (Rhetorical Structure Theory; RST) [9] では，談話構造を句構造木 (**修辞構造木**) で表す．Carlson らによって RST に基づいた大規模なコーパス (RST Discourse Treebank; RST-DTB) が構築されて以降 [1]，言語処理分野において，RST の応用は大きく進展した．一方，文書要約や参照解析のようなタスクでは，句構造よりもテキストスパン同士の親子関係を直接表せる依存構造の方が望ましいが [2, 5]，談話依存構造木をアノテーションしたコーパスは存在しないため，平尾ら [5] は RST-DTB の修辞構造木を談話依存構造木へと自動変換する手法を提案した．また，Li ら [7] も同様の変換法を独立に提案したが，平尾らの変換法とは異なる原理で動作するため，2 つの手法から生成される談話依存構造木には本質的な違いがある．

本稿では Li らと平尾らの木変換法の違いを明らかにするため，次の 3 つのことを行った．

(1) 2 つの変換法を統一的なアルゴリズムで記述し，その動作原理の違いを明らかにする．
(2) 2 つの談話依存構造木の構造的な性質の違いを明らかにする．
(3) RST とは異なるタイプの談話構造の表現である Penn Discourse Treebank (PDTB) [11] で用いられる 2 つのテキストスパンとそれらを結ぶ関係 (談話における一種の述語項構造関係) の解析への有用性を議論する．

依存構造の利点の 1 つは，統語表現として述語項構造に基づく意味構造と親和性が高いことであり，(3) の実験結果から，(特に平尾らの) 談話依存構造木は PDTB の述語項構造との間に高い関連性を示した．

2 修辞構造木

修辞構造木において，その葉ノードは **Elementary Discourse Units** (EDUs) と呼ばれる文や節等の談話の基本単位に対応する．隣接する EDU 同士は**修辞関係**によって結合し，より大きな**談話単位**を形成する．さらに，その談話単位同士も修辞関係によって結合され，木構造を形作る．図 1 に RST-DTB の修辞構造木の例を示す (wsj0623)．各 EDU は次のテキストに対応する．

$\Big\{$[The fiscal 1989 budget deficit figure came out Friday .]$_{\text{e-1}}\Big\}_1$, $\Big\{$[It was down a little .]$_{\text{e-2}}\Big\}_2$, $\Big\{$[The next time you hear a Member of Congress moan about the deficit ,]$_{\text{e-3}}$, [consider what Congress did Friday .]$_{\text{e-4}}\Big\}_3$, $\Big\{$[The Senate , 84-6 , voted to increase to $ 124,000 the ceiling on insured mortgages from the FHA ,]$_{\text{e-5}}$, [which lost $ 4.2 billion in loan defaults last year .]$_{\text{e-6}}\Big\}_4$, $\Big\{$[Then , by voice vote , the Senate voted a pork-barrel bill ,]$_{\text{e-7}}$, [approved Thursday by the House ,]$_{\text{e-8}}$, [for domestic military construction .]$_{\text{e-9}}\Big\}_5$, $\Big\{$[the Bush request to what the Senators gave themselves :]$_{\text{e-10}}\Big\}_6$, . . .

ここで [] の添字にテキスト内の EDU の ID，{} の添字に文の ID を書いた．

ある談話単位は，その修辞関係上において他の談話単位よりも重要な情報を持つ場合，**核** (N)，そうでない場合，**衛星** (S) と呼ばれる状態で特徴付けられる．修辞関係は，**単一核** (mononuclear)，または，多核 (multinuclear) の関係に分けられ，単一核の修辞関係は核と衛星の談話単位を結び，多核の修辞関係は 2 つ以上の核となる談話単位を結ぶ．さらに，修辞関係には 'Elaboration' や 'Condition' などのより詳細に関係を記述する型が与えられる．図 1 では，例えば，e-3 の親である衛星と e-4 の親である核の談話単位が 'Elaboration' という型を持つ単一核の修辞関係で結ばれている．また，e-5,e-6 を覆う核と e-7,e-8,e-9 を覆う核の談話単位は 'Temporal' という型を持つ多核の修辞関係で結ばれている．

図 1 からも分かるように，修辞構造木は文脈自由文法 (CFG) で記述された一種の句構造木として見ることができる．EDU は終端ノード，談話単位は非終端ノードに対応し，修辞関係は CFG 規則のようにエッジを形成する．次節では，この修辞構造木を談話依存構造木へと変換する手法について説明する．

Algorithm 1 convert-RST-DT-to-DDT

Require: RST discourse tree: $\boldsymbol{t}$
Ensure: Discourse dependency tree: $\boldsymbol{dt}$
1: $\boldsymbol{dt} \leftarrow \emptyset$
2: **for all** EDU e-j in $\boldsymbol{t}$ **do**
3: $P \leftarrow \begin{cases} \text{find-My-Top-Node(e-}j\text{)} \text{ // Li14} \\ \text{find-Nearest-S-Ancester(e-}j\text{)} \text{ // 平尾 13} \end{cases}$
4: **if** isRoot(P) = TRUE **then**
5: $\ell \leftarrow$ Root
6: $i \leftarrow 0$
7: **else**
8: $\ell \leftarrow$ Label(P)
9: $P' \leftarrow$ Parent(P)
10: $i \leftarrow$ find-Head-EDU(P')
11: **end if**
12: $j \leftarrow$ Index(e-j)
13: $\boldsymbol{dt} \leftarrow \boldsymbol{dt} \cup (i, \ell, j)$
14: **end for**
15: Return $\boldsymbol{dt}$

3 談話依存構造木への変換法

3.1 Li14 の変換法

Li ら [7] の変換法は Penn Treebank の句構造木に対する主辞割当規則 [8] と同様の考え方に基づく．修辞構造木上での主辞とは，ある談話単位の構文的な機能

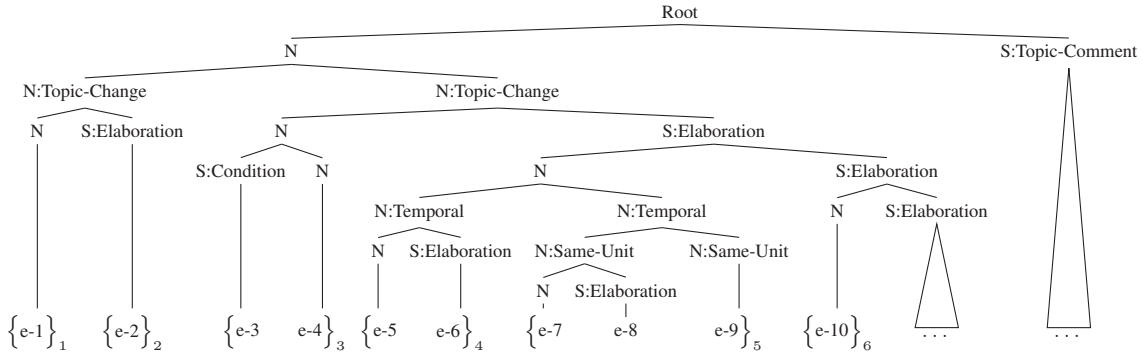

図 1: 修辞構造木の例 (wsj0623): ‘S’，‘N’，‘e’ は衛星，核，EDU を表す．{} で同じ文内の EDU をまとめている．

Algorithm 2 find-My-Top-Node(e)

Require: EDU: e
Ensure: C
1: $C \leftarrow e$
2: $P \leftarrow$ Parent(e)
3: **while** LeftmostNucleusChild(P) $= C$ and isRoot(P) $=$ FALSE **do**
4: $C \leftarrow P$
5: $P \leftarrow$ Parent(P)
6: **end while**
7: **if** isRoot(P) $=$ TRUE **then**
8: $C \leftarrow P$
9: **end if**
10: Return C

Algorithm 3 find-Head-EDU(P)

Require: non-terminal node: P
Ensure: i
1: **while** isLeaf(P) $=$ FALSE **do**
2: $P \leftarrow$ LeftmostNucleusChild(P)
3: **end while**
4: $i \leftarrow$ Index(P)
5: Return i

を決定する EDU のこととする．Li らの変換法では，修辞構造木のノードをボトムアップに訪れ，次のような基準で各ノードに主辞を割り当てる: 最左の核をとる子ノードの主辞を主辞とする; もし子ノードに核が無ければ，最左のノードの主辞を主辞とする．この主辞割当規則は Sagae の研究 [12] において，初めて提唱されたものである．

Li らの変換法では，右分岐に二分木化した修辞構造木 [4] に上の規則で主辞を割り当てた後，その木を談話依存構造木へと変換する．ここでは Algorithm 1-3 にこの変換法を示す．ただし，次節で説明する平尾 13 の変換法との違いを簡潔に記述するため，上で説明した手順とは異なる形でアルゴリズムを記述した．Algorithm 1 に示すように，メイン関数は与えられた修辞構造木 $\boldsymbol{t}$ の各 EDU に対して，その親となる EDU を直接探す．この手順は大きく分けて 3 つである．

まず, Algorithm 1 の 3 行目で Algorithm 2 を呼び出し，先の主辞割当規則の基準において，現在処理されている EDU e-j が主辞として割当られる最も高い位置の非終端ノードを見つける．ここで，Parent(P) はノード P の親ノードを，LeftmostNucleusChild(P) は最左の核となる子ノードを返す関数である．

次に, Algorithm 2 がノード P を返したとき，Algorithm 1 では P の親ノードに割り当てられる主辞を探す．P が t のルートである場合，型 ℓ を ‘Root’，e-j の親となる EDU の位置番号 i を特殊な仮想 EDU e-0 の位置 0 に設定する (Algorithm 1 の 5-6 行). そうでない場合，$\ell \leftarrow$ Label(P)，$P' \leftarrow$ Parent(P) と設定する (Algorithm 1 の 8-9 行)．ここで Label(P) はノード P に割り当てられた修辞関係の型を返す関数である．もし，P が型を持たない場合，特殊な型 ‘Span’ を返す．次に，Algorithm 3 を呼び出し，P' から始めて，最左の核となる子ノードへの移動を終端ノード e-i に到達するまで繰り返す．そして，関数 Index(P) によって，発見した EDU の位置番号 i を返す．

最後に，修辞関係の型 ℓ で親 e-i に e-j をかけた依存関係の辺を作る．この辺は (i,ℓ,j) として書く．

図 2 は図 1 の修辞構造木に対して，Li らの変換法を適用してできる談話依存構造木を表す．各 EDU には，修辞構造木上でそれらの親ノードに割り当てられた ‘N’ か ‘S’ の状態を与えた. 図 2 のように，Li らの変換法では交差なしの談話依存構造木が必ず作られる．

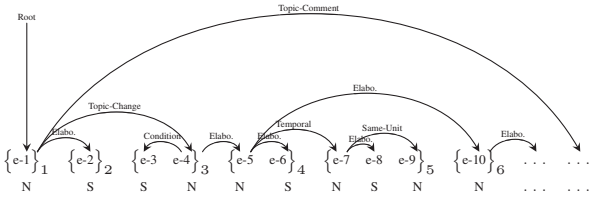

図 2: 図 1 の修辞構造木に Li14 の変換法を適用したときにできる談話依存構造木.

Algorithm 4 find-Nearest-S-Ancester(e)

Require: EDU: e
Ensure: P
1: $P \leftarrow$ Parent(e)
2: **while** isNucleus(P) $=$ TRUE and isRoot(P) $=$ FALSE **do**
3: $P \leftarrow$ Parent(P)
4: **end while**
5: Return P

3.2 平尾 13 の変換法

平尾らの変換法 [5] は，Li14 の変換法とは異なるが，その違いは，各 EDU が主辞として割り当てられる最も高い位置のノードが異なることだけである．平尾らの変換法では，Algorithm 1 の 3 行目で Algorithm 4 を呼び出し，ある EDU に対して，そこからルートまでの経路上で，直近にある衛星のノードを返す．このように直近の衛星が主辞として割り当てられる最も高い位置のノードとなる基準は，Veins Theory [2] の主辞割当規則と本質的に等価である．

図 3 には図 1 の修辞構造木に平尾らの変換法を適用

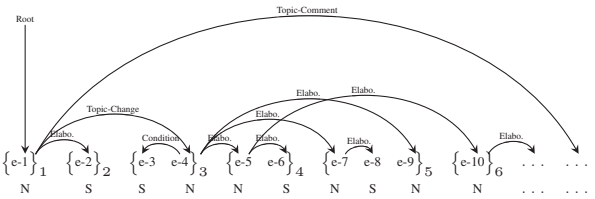

図 3: 図 1 の修辞構造木に平尾 13 の変換法を適用したときにできる談話依存構造木.

表 1: 談話依存構造木の辺に付けられた型ラベルの分布.

label	Li14	平尾 13
Attribution	3070	3182
Background	937	1176
Cause	692	731
Comparison	300	200
Condition	328	344
Contrast	1130	838
Elaboration	7902	10358
Enablement	568	609
Evaluation	419	596
Explanation	986	1527
Joint	1990	42
Manner-Means	226	272
Root	385	385
Same-Unit	1404	62
Span	1	0
Summary	223	332
Temporal	530	271
TextualOrganization	157	137
Topic-Change	205	401
Topic-Comment	336	326

したときにできる談話依存構造木を示した．Li14 の変換法とは異なり，平尾らの変換法では談話依存構造木に交差が生じる．単一核の修辞関係から作られる依存関係の辺は両手法とも同一であるが，多核の修辞関係から作られる辺が異なるため，生成する木に違いが出る．例えば，図 1 の文 4 と文 5 は ‘Temporal’ の型を持った多核の修辞関係を持つ．Li14 の変換法から作られた談話依存構造木 (図 2) では，その 2 文は ‘親子’ の依存関係を持つが，平尾らの変換法では ‘兄弟’ の依存関係を作る (図 3)．後述するように，平尾らの変換法で作られる談話依存構造木は，PDTB の述語項構造を構文的により良く説明できるが，Li14 の変換法よりも複雑な木構造を作り出す．

4 実験

4.1 談話依存構造木の分析

4.1.1 型ラベルの分布

実験では RST Discourse Treebank (RST-DTB) を用いた．RST-DTB は英語の Wall Street Journal (wsj) の 385 文書に修辞構造木をアノテーションしたコーパスである．実験では慣習に従い，18 種類の修辞関係の型ラベルを使い，Li14，及び，平尾 13 の変換法を使って，談話依存構造木へと変換した．

表 1 には，談話依存構造木の辺に付与された 18 種類の型ラベルと 2 種類の特殊な型ラベル ‘Span’, ‘Root’ の分布を示した[1]．ここでは，下線を引いた 3 つの型

表 2: 談話依存構造木の構造的な性質の統計量.

property	Li14	平尾 13
最長経路長	10.2	8.4
ノード数 (深さ 2, 3, 4)	6.5, 14.3, 23.3	9.6, 22.1, 35.0
gap degree 0, 1, 2	385, 0, 0	113, 260, 12
projective	385	113
well-nested	385	385

に議論を絞る．平尾 13 の変換法では，‘Elaboration’ ラベルが大きく増加し，‘Joint’ と ‘Same-Unit’ ラベル (多核の修辞関係に付与される型) が大きく減少している．これは Algorithm 4 において直近の衛星を探すときに，その経路上に現れる多核の修辞関係は全て無視され，衛星に与えられた型 (多くの場合，‘Elaboration’) が辺ラベルとなるからである．平尾 13 の変換法において，特に ‘Elaboration’ となる型ラベルに捨てられた多核の修辞関係の情報を残し，ラベルを詳細化することは今後の重要な課題である．

4.1.2 構造の複雑さ

表 2 では談話依存構造木の構造的な性質を調べた．まず，仮想ルート e-0 から末端の EDU までの最長経路長の平均，e-0 から深さ x までのノード数の平均を示した (e-0 は深さ 0)．結果から，平尾 13 の変換法で作られる談話依存構造木はより幅が広く，浅い構造になっていることがわかる．節 3.2 で述べたように，これは多核の修辞関係から作られる依存関係が，‘親子’ あるいは ‘兄弟’ となる違いに由来するものである．

表 2 ではまた，無交差 (**projectivity**)，**gap degree**, **well-nestedness** と呼ばれる依存構造グラフの制約を，どれだけの談話依存構造木が満たしているかを調べた．これらの詳細な定義は文献 [6] を参考にしてもらいたい[2]．Li14 の変換法では全ての依存構造木が無交差であった．一方，gap degree が高い場合，依存構造木は複雑な交差を含んだ構造となるため，平尾 13 の変換法で作られる多くの依存構造木は交差を含むことがわかる．しかし，それらは全て well-nested な制約を満たしており，多くは gap degree が 1 程度である．交差なしや well-nested で gap degree が低い依存構造木に対しては効率的な多項式時間の構文解析アルゴリズムが知られている [3].

4.2 談話レベルの述語項構造との比較

PDTB には 2 つのテキストスパンを項，それらに対する 1 つの談話結合子を述語とした述語項構造がアノテーションされている．その結合子が持つ意味を与えられる項は Arg2，もう一方は Arg1 と呼ばれ，結合子は Explicit，Implicit，AltLex，EntRel，NoRel の 5 種類に分類され，さらに Explicit，Implicit，AltLex にはより詳細な型が与えられる．wsj の 364 文書は PDTB 2.0 と RST-DTB 両方のアノテーションを持つ．例えば，wsj0623 文書は次のような述語項構造を持つ:

[The next time you hear a Member of Congress moan about the deficit , consider what Congress did Friday .]$_{\text{Arg1}}$ [The Senate , 84-6 , voted to increase to $ 124,000 the ceiling on insured mortgages from the FHA , which lost $ 4.2 billion in loan defaults last year .]$_{\text{Arg2}}$

[1] 修辞構造理論に従えば，Li14，及び，平尾 13 の変換法共に ‘Span’ ラベルが辺に割当てられることはない．表 1 において，‘Span’ ラベルが生じたのは，wsj1189 ファイルの修辞構造木の e-7 から e-9 を覆う部分木のアノテーションエラーによるものと考えられる．

[2] 文献 [6] とは異なり，仮想ルート e-0 における辺 $(0, \text{Root}, i)$ も含めて計算を行っている．

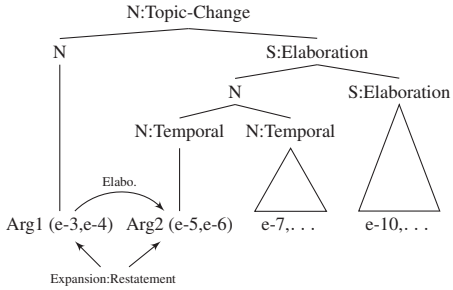

図 4: 述語項構造を埋め込んだ修辞構造木.

表 3: 述語項構造の項ペアが談話依存構造木上で ‘親子’ または ‘兄弟’ の依存関係となる数.

	親子	兄弟	それ以外
Li14	4462	714	608
平尾 13	3437	1656	691

(Implicit Expansion:Restatement)

ここでは結合子に ‘Expansion:Restatement’ という型が与えられている．PDTB のそれら 364 文書にはこのような述語項構造が合計 7026 個付与されている．

談話依存構造と PDTB の述語項構造との関係を調べるため，それらのアノテーションの融合を試みた．まず，PDTB の項がとるテキストスパンは，RST-DTB における EDU のスパンと必ずしも一致しないため，PDTB における項のテキストスパンを完全に覆うことができる最小の EDU 系列を求め，修辞構造木上でその EDU 系列の代わりに項を埋め込んだ．図 4 には図 1 で示した修辞構造木の一部を示しており，上で例示した述語項構造が埋め込まれている．実験では 7026 個のうち 1242 個の述語項構造については修辞構造木に埋め込むことができなかった．これは，項のテキストスパンが不連続であったり，修辞構造木の句構造と不整合であったからである．

次に，埋め込みに成功した 5784 個の修辞構造木を談話依存構造木へと変換した．図 4 のように，Li14 と平尾 13 の変換法ともに依存関係の辺 (Arg1, Elaboration, Arg2) を作るが，これからは談話依存構造木の方が修辞構造木よりも直接的に述語項構造との関係を表せていることがわかる．表 3 では，談話依存構造木上で ‘親子’ または ‘兄弟’ の依存関係となる項ペアの数を調べた．5784 個のうち約 90%の述語項構造が，それらの関係を持ち，異なるアノテーション基準で作られた 2 つのコーパスの間に強い関連性があることが明らかとなった．

平尾 13 の談話依存構造木では，‘兄弟’ の依存関係を持つ項ペアが増加している．これらの増加した項ペアのほとんどが ‘Expansion:Conjunction’ や ‘Expansion:List’ の型を持つ結合子で結ばれていた．大まかに言って，これらの型は 2 つの項に記述されたテキストスパンが等価な情報を持つ場合に使われる．文内の並列構造を依存構造木で表す場合も議論 [10] があるように，このような並列関係を ‘親子’ または ‘兄弟’ のどちらで表すのが良いかは一概には決められない．しかし，図 5 のような形で複数の ‘Expansion:List’ を型に持つ述語項構造が現れるとき，これらの項は全て互いに等価な関係にある．Li14 の ‘親子’ による表現方法では，それらの等価性を依存構造木の形から判断することは難しいが，平尾 13 の ‘兄弟’ による表現方法ではそれらの等価な項は全て同列に表現されるため，

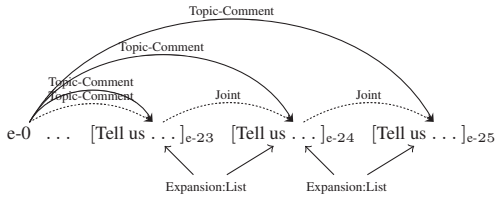

図 5: ‘Expansion:List’ の型を持つ 2 つの述語項構造に対する平尾 13 (実線) と Li14 (点線) の依存構造 (wsj0623).

構造による判断が容易になる．

5 おわりに

本稿では Li14 と平尾 13 の談話依存構造木を比較し，その性質の違いを明らかにした．また，それらの談話依存構造木を用いて，RST-DTB と PDTB という 2 つの異なる基準でアノテーションされたコーパスの間に強い関連性があることを初めて示した．今後は，談話依存構造木を PDTB 形式の談話構造解析に活かすことを検討している．

謝辞

本研究の一部は，JSPS 科研費 基盤研究 (B) 26280079，若手研究 (B) 26730126 の助成をそれぞれ受けたものである．

参考文献

[1] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. *Building a discourse-tagged corpus in the framework of rhetorical structure theory*. Springer, 2003.

[2] Dan Cristea, Nancy Ide, and Laurent Romary. Veins theory: A model of global discourse cohesion and coherence. In *Proceedings of the 17th COLING*, pp. 281–285, 1998.

[3] Carlos Gómez-Rodrıguez, David Weir, and John Carroll. Parsing mildly non-projective dependency structures. In *Proceedings of the 12th EACL*, pp. 291–299, 2009.

[4] Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. Hilda: a discourse parser using support vector machine classification. *Dialogue & Discourse*, Vol. 1, No. 3, 2010.

[5] Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. Single-document summarization as a tree knapsack problem. In *Proceedings of the EMNLP*, pp. 1515–1520, 2013.

[6] Marco Kuhlmann and Joakim Nivre. Mildly non-projective dependency structures. In *Proceedings of the COLING/ACL*, pp. 507–514, 2006.

[7] Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. Text-level discourse dependency parsing. In *Proceedings of the 52nd ACL*, pp. 25–35, 2014.

[8] David M Magerman. *Natural language parsing as statistical pattern recognition*. PhD thesis, University of Pennsylvania, 1994.

[9] William Mann and Sandra Thompson. Rhetorical structure theory: Towards a functional theory of text organization. *Text*, Vol. 8, No. 3, pp. 243–281, 1988.

[10] Martin Popel, David Marecek, Jan Stepánek, Daniel Zeman, and Zdenek Zabokrtskỳ. Coordination structures in dependency treebanks. In *Proceedings of the 51st ACL*, pp. 517–527, 2013.

[11] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. The penn discourse treebank 2.0. In *LREC*, 2008.

[12] Kenji Sagae. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proceedings of the 11th IWPT*, pp. 81–84, 2009.