プロットの作り方 v3

塩谷 亮太

はじめに

- プロットとは：
  - 話の筋を箇条書きの形でまとめたもの
  - 文章や発表スライドを書く前に，まずプロットを作る必要がある
- なぜプロットを作るのか：
  - 話の筋を整理し，その筋に収束するように全体を構成する
  - そうすることで，主張を明確に示すことができる
    - そうしないと，「言いたいことがなんとなく適当に並べられた良くわからないもの」が出来上がる

プロットのタイプ

- 詳細度ごとに複数のプロットを以降では説明
  - １点プロット（=目標規定文）
  - ３点プロット
  - ６点プロット
  - 完全プロット
- N点プロット = N 個の項目から成るプロット
  - 点数が多いほど分量が増え詳細になる

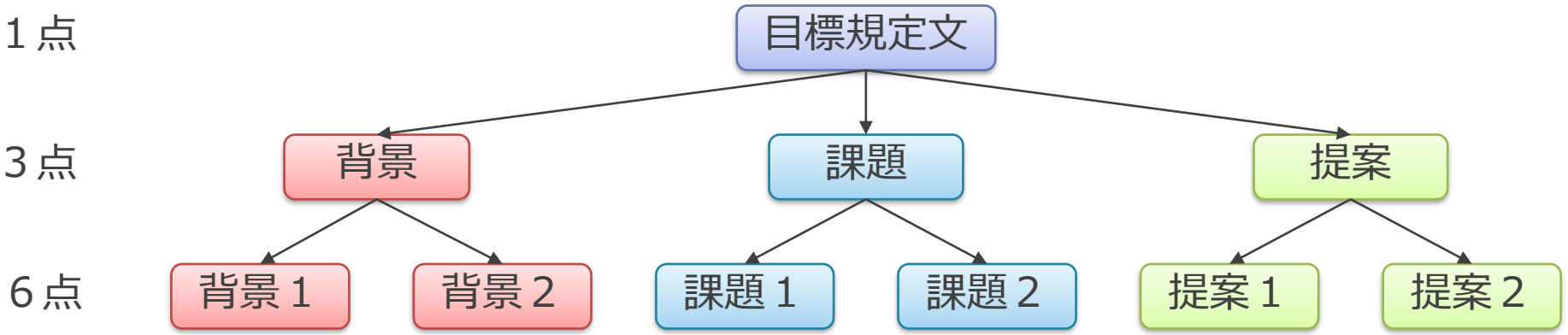
まず３点プロットから始める

- １点プロットを最初に作るのはかなり難しい
  - ◇ 本当に大事な事だけを１つの文に集約/圧縮する必要がある
  - ◇ しかし，何が真に大事なのかは最初はわからない
- ６点プロットや完全プロットも難しい
  - ◇ 自由度が高すぎてまとめるのが難しい
- ３点プロットが規模的にちょうどよい

３点プロットは最初に作るのにちょうどよい

- ３点プロットはいわば「プロットのプロット」
  - ◇ まず「背景→課題→提案」の流れの各要素が何なのかをはっきりさせる
  - ◇ これを元に，６点プロットや完全プロットを作る
- 規模が小さくかつ形式が決まっているので，考えやすい
  - ◇ スライド１枚程度にまとまる
  - ◇ 短いので，まず取っ掛かりとして始めやすい
- ３点プロットに実際に取り掛かる前に，この資料は最後まで読んでほしい
  - ◇ 最終的に完全プロットを作るところまでの道筋を意識してほしい

詳細度と論理構造



- ツリーの上下は説明の詳細度に対応している
  - ◇ 上の階層（点数が少ない方）は，下の階層の要約になっている
  - ◇ 下の階層は上の階層をより詳しく述べている
- ３点プロットから初めて，
  - ◇ 上に登る（まとめる）ことや，
  - ◇ 下に降りる（詳細を肉付けする）していく

３点プロット

３点プロット：この３点で話の筋をまとめる

1. 背景：主張全体の背景や問題を説明する
   - ◇ 全体として解決しようとしている問題の説明
2. 課題：解決しようとしている課題を説明する
   - ◇ 背景となる問題に対する既存手法の説明とその問題点
   - ◇ 既存手法がない場合は，背景の中で着目する問題を掘り下げる
3. 提案：課題であげられた問題を解決する提案手法を説明する
   - ◇ 課題に対する洞察や観察
   - ◇ キーとなるアイデア
   - ◇ なぜ課題を解決できるのか

例１：小田喜くんの輪講の例 = 既存手法があるパターン
（輪講なので具体的なアイデアがまだない事に注意）

- 背景：SIMT アーキテクチャにおける冗長な演算
  - ◇ SIMT(D) では基本的には複数のデータに対して同じ演算を行う
  - ◇ しかし SIMT では全く同じ冗長な演算を複数のレーンで行っている場合があり無駄
- 課題：スカラ化とその問題
  - ◇ 冗長な演算を１つの演算にまとめるスカラ化が提案されている
  - ◇ 従来のスカラ化では制約があり効果的にまとめられない
- 提案：スカラ化の改良
  - ◇ Temporal SIMT と動的なスカラ化の組み合わせにより実現

例２：小泉くんの DATE = 既存手法があるパターン

- 背景：早いプリフェッチによるレイテンシの隠蔽
  - プリフェッチ：キャッシュにデータを先読みしておく技術
  - 多くの研究では十分に早くプリフェッチすることを重視
    - メモリ・アクセスのレイテンシを有効に隠蔽するため
    - 通常はなるべく遠い未来のアドレスを予測してプリフェッチ
- 課題：早すぎるプリフェッチ
  - 早すぎると使用される前にキャッシュから追い出される
  - これにより性能向上の機会が大きく失われている
- 提案：プリフェッチを遅らせる
  - データが参照されるタイミングまであえて遅らせる

例３：木村さんの輪講 = 既存手法がないパターン
（輪講なので具体的なアイデアがまだない事に注意）

- 背景：ベクトル命令
  - ◇ 単一の命令で可変長の複数データを処理する命令の方式
  - ◇ データ並列性のある処理を対象
  - ◇ RISC-V ベクトル拡張などの形で実装されている
- 課題：ベクトル命令の実装コスト
  - ◇ ベクトル命令では１つの命令が多数のアクセスを発生させる
  - ◇ 従来の作り方で out-of-order プロセッサ上に実装すると，複雑さが爆発する
- 提案：ベクトル命令の実装の複雑さを下げる

例４：出川くんの ICCD = 既存手法がないパターン

- 背景：命令キャッシュ・ミス数を使った性能の見積もり
  - ◇ 命令キャッシュに関わる研究ではミス数が主要な評価項目だった
  - ◇ ミス数が減ると基本的には実行時間が短くなるため
- 課題：シミュレーション時間
  - ◇ 現代のプロセッサではミス数と実行時間が直接相関しない
  - ◇ 精度よい性能見積もりのためにはプロセッサ全体のシミュレーションが必要
  - ◇ しかしシミュレーションには長い時間かかる
- 提案：命令キャッシュ・ミス数に代わる新たな指針
  - ◇ その指針を使った高速な性能見積もりの提案
  - ◇ ２桁短い時間でシミュレーションとほぼ同じ精度の性能見積もりを実現

とりあえずこのフォーマットにまとめる事を目指す

1. 背景：「～」= 背景を一つの文ないしは名詞でまとめる
   - ◇ 「～」= 背景を説明する１つの文
   - ◇ 「～」・・・
2. 課題：「～」
   - ◇ 「～」
   - ◇ 「～」
3. 提案：「～」
   - ◇ 「～」
   - ◇ 「～」

プロットを作る際の実際の手順

- 以下の手順で進めると，作りやすい：
  1. まずは思いつく関連しそうな項目をたくさん書き出してみる
     - 名詞や短文の形にして並べてみる
  2. 各項目を整理
     1. 関係する項目同士をくくって親子にまとめる
        - まとめた項目を一言で表したまとめの短文（親）を作る
        - まとめの短文の下に子項目としてインデントして置く
     2. 内容が冗長なものをマージしたり削除する
  3. ある程度まとまったら，それらを背景，課題，提案に分類する
     - このとき，次のページで説明する関連を意識する
     - うまく分類して関係を説明できない時は，本質的に関係ない可能性がある

項目間の関係

- 「課題」に書く内容は，「背景」で提示した問題に対応させる
  1. 「背景」で提示した問題を解決する
     - 既存手法がある場合，基本的にはこの形になる
  2. 「背景」の特定の問題に着目して掘り下げる
     - 特に既存手法がない場合，こちらになることもある

- 「提案」では，「背景」と「課題」で提示した問題に対応させる
  - 基本的には「課題」で提示した問題を解決する方法を書く
  - そうすれば，普通は自然と「背景」の問題にも対応する

- 「背景→課題→提案」の流れが明確にわかるようにする
  - これらに直接つながらない事は，入れてはいけない

箇条書きを作る際の形式上の注意

- 複文を使うのは原則禁止
  - ◇ 複数の文を繋げて１つの文にしてしまうと，関係が良くわからなくなりがち
  - ◇ 最初は単文のみで作る
    - □ どうしても複文を入れる場合は，１行に収まる長さまで
    - □ ３文以上からなる複文は常に禁止
- 同様の理由により，長い修飾節を持った文も使わない
  - ◇ 長くなってしまう場合は，インデントをしてぶら下げると良い
- 一つの項目にぶら下げる項目は３つまで
  - ◇ ４つ以上の項目が同じレベルに並んでいる場合は，それらをインデントにまとめる
  - ◇ ３つより多いと関係が曖昧になる

応用：４点プロット

- 洞察を加えた４点プロットの形でもよい
  - 背景
  - 課題
  - 洞察
    - 課題に対する新しい観察結果や，課題の核心の新しい解釈など
  - 提案
- 洞察は課題や提案の下にぶら下がることもある
  - 小泉くんの例での「早すぎるプリフェッチ」は洞察でもある

３点プロットのチェック・リスト

- 以下が満たされているかを確認：

1. 背景，課題，提案の３項目から成っているか？
2. 課題は背景の問題に，提案は課題の問題に対応しているか？

3. 複文を含んでいないか？
4. １行を越えるような長い修飾節を含んだ文が入っていないか？
5. ４つ以上の項目を並列に並べていないか？

１点プロット

１点プロット = 目標規定文

- １点プロット = 目標規定文
  - その文章の主張を１つの文の形にまとめたもの
    - 目標規定文は「理科系の作文技術」より
  - もっとも短い形のプロットとも言える

- 「理科系の作文技術」の説明
  - 「自分は何を目標としてその文章を書くのか，そこで何を主張しようとするのかを熟考して，それを一つの文にまとめて書いてみることを勧める」
  - 「主題に関してあることを主張し，または否定しようとする意思を明示した文」（コーベットによる thesis の説明）

目標規定文

- 目標からトップダウンに構成を作る
  - 目標規定文を作り，その目標に収束するように文章全体の構想を練る
  - この目標に繋がる内容のみを全体の構成に残す
    - 「関係はしているが，あってもなくても良い」みたいなものは入れてはいけない
- 主張全体を論理的なツリーとして表した際のルートにあたる
  - これをさらに短くまとめたものがタイトルになる

３点プロットと目標規定文の関係

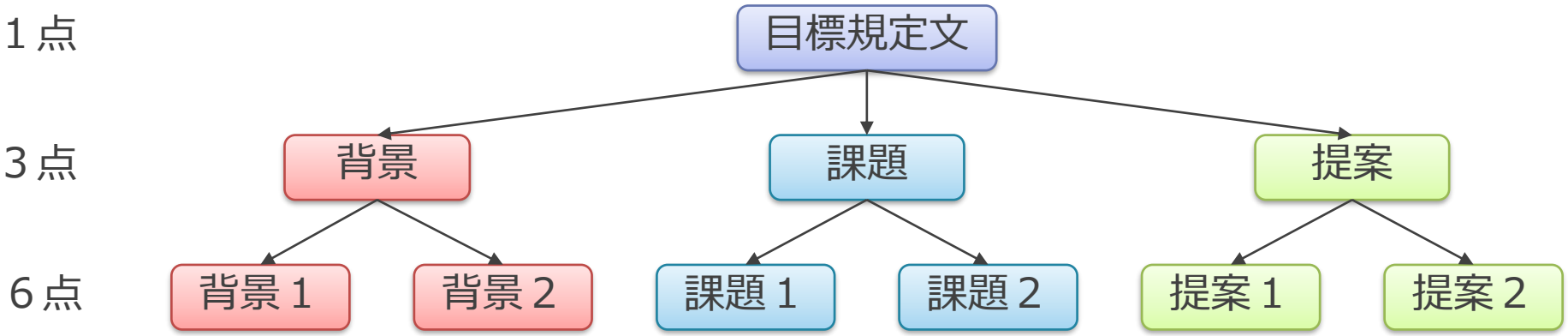
- 作り方：
  - ３点プロットが出来たら，そこからさらに真に重要な項目を抽出
  - それらを繋げて目標規定文にする

6点プロット

６点プロット

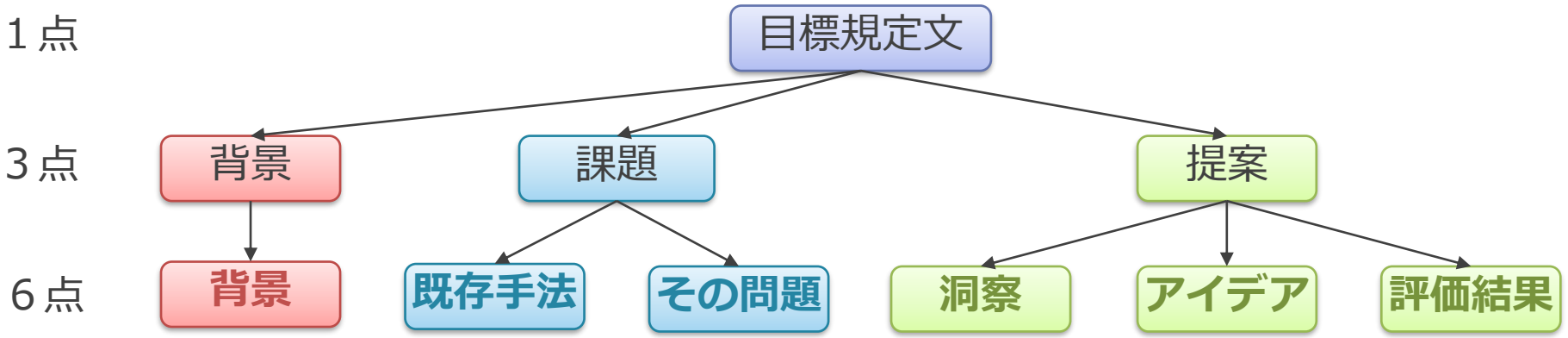
- ６点プロットはイントロを書く際に作る
  - 論文のイントロは典型的には６パラグラフ前後
  - この各パラグラフで何を話すかをまとめる
    - ６点プロットの各項目は各パラグラフのトピックセンテンスに対応する
    - トピックセンテンスだけを繋げて読んでも意味が通るように
- ６点プロットも３点プロットから派生させて作る

詳細度と論理構造



- ツリーの上下は説明の詳細度に対応している
  - ◇ 上の階層（点数が少ない方）は，下の階層の要約になっている
  - ◇ 下の階層は上の階層をより詳しく述べている
- ３点プロットから初めて，論理ツリーの関係が保たれているかをチェックする

詳細度と論理構造



- 6点プロット時の配分には自由度がある
  - ◇ 3点のどこをどれだけ詳しく話すかは，話題による
  - ◇ 基本的には背景は1つのままで，提案を増やす事が多い

完全プロット

完全プロット

- 論文やスライド全体のプロット
  - 特に項目数などの形式はない：
    - 基本的には３点プロットから派生させて考える
    - ６点プロットでは省略されるような実装の詳細なども入る
- ３点プロットは，いわば「完全プロットのプロット」
  - ３点プロットで整理した内容をもとに，完全プロットを作る
  - いきなり完全プロットを作るのは難しい

完全プロット

- 論文用：以下を箇条書きにまとめる
  - 論文の subsubsection までの節タイトル
  - そこで何を話すか
- スライド用：以下を箇条書きにまとめる
  - スライドの各ページのタイトル
  - そこで何を話すか

スライド用のプロットの構成の例

1. イントロ
2. 背景となる問題
3. 既存手法
    1. 既存手法の説明
    2. 既存手法の問題
4. 提案手法
    1. 構成，動作，例
    2. 既存手法との比較
5. 評価
6. まとめ