

プロットの作り方 v6

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
 - いわば文章やスライドの設計図にあたるもの

■ 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
 - 設計図なしで建物を建てるとヒドい事になるのと同じ

プロットのタイプ

- 詳細度ごとに複数のタイプ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 (イントロ用)
 - ◇ 完全プロット (論文/スライド全体用)
- 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
- とりあえずどう言うものなのかは, 9 ページ目以降の例を参照

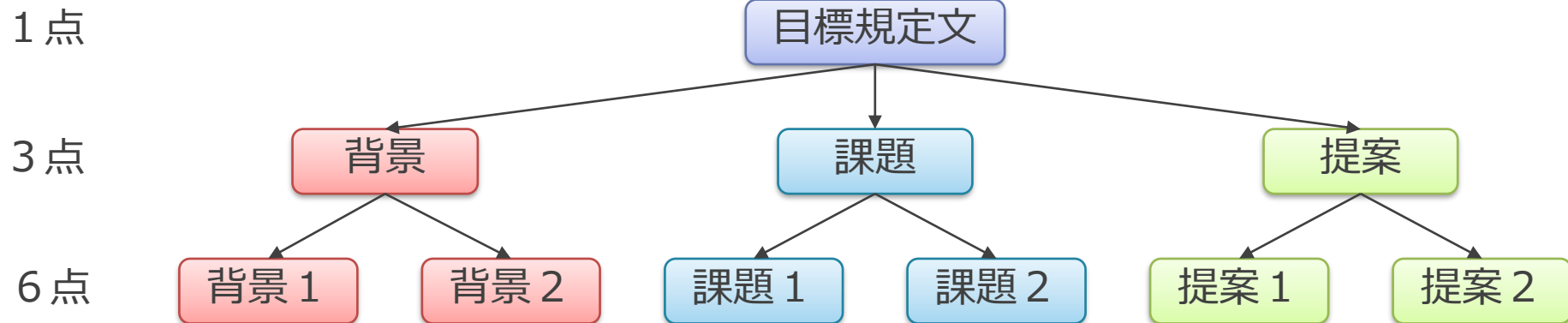
まず 3 点プロットから始める

- 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、内容を膨らませて6点/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（=概要にまとめる）ことや,
 - ◇ 下に降りる（=詳細を肉付けする）していく

全体の目次

1. プロットの作り方
 1. 3点プロット
 2. 1点プロット
 3. 6点プロット
 4. 完全プロット
2. プロットから文章へ

3点プロット

3点プロットの目次

1. 3点プロットとは
2. 作り方
 1. 内容のまとめかた
 2. 箇条書きを作る上での形式的なポイント
3. チェック・リスト

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
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ます無駄

■ 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
- ◇ しかし, 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関連しない
 - ◇ 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

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点プロットの目次

1. 3点プロットとは
2. 作り方
 1. 内容のまとめかた
 2. 箇条書きを作る上でのポイント
3. チェック・リスト

この形式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：「～」

◇ 「～」

◇ 「～」

以下の手順で進めると、作やすい

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

- それらを一言で表した、まとめの短文（親）を作る

- 親の下に、それらを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

- 「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
- 「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. ないしは、「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

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し、理解しづらい

箇条書きの親子関係

- $A \rightarrow B \rightarrow C$ のような演繹の関係を下のような箇条書きにしていまいがちだが、これは良くない

- ◇ (親子関係は基本的に 概要 \leftrightarrow 詳細 を表すもの)

- ◇ A : ~

- B : ~

- * C : ~

- このような場合は $A \rightarrow B \rightarrow C$ の主張を一言にまとめたものを (X) を作り、その下にぶらさげる

- ◇ X :

- A : ~

- B : ~

- C : ~

演繹の関係にある要素の書き換えの例

$A \rightarrow B \rightarrow C$ を X の下に展開

- A : 人間の脳の一時記憶の大きさには限りがある
 - ◇ B : このため, 人間は 5 ~ 7 個以上の事柄を一度に把握できない
 - C : したがって, 余裕を持って 3 個程度以内にするのがよい

- X : 1 項目にぶら下げる項目の数は少なめにする
 - ◇ A : 人間の脳の一時記憶の大きさには限りがある
 - ◇ B : このため, 人間は 5 ~ 7 個以上の事柄を一度に把握できない
 - ◇ C : したがって, 余裕を持って 3 個程度以内にするのがよい

なぜ親子関係のある箇条書き（階層構造）にまとめるのか？

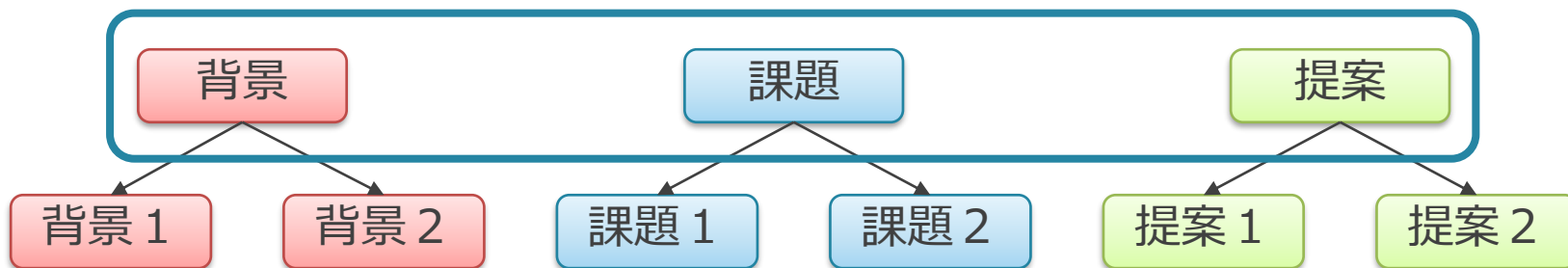
- すべての項目がフラットに並んでいると内容の把握が難しい
 - ◇ 人間の頭が一度に扱える量には限界がある
 - ◇ 5個ぐらいからは並列に並んでいると厳しくなってくる
- 階層化して一度に考えることの数減らす
 - ◇ 関係ある項目ごとに要約にまとめて階層化
 - ◇ 階層化すると、一度に考えることの数減る
- 一度に考える話題の数が小さいままに、以下が両立できる：
 - ◇ 話の大筋をつかむ
 - ◇ 各部分の詳細を理解する

一度に考える必要がある話題の数

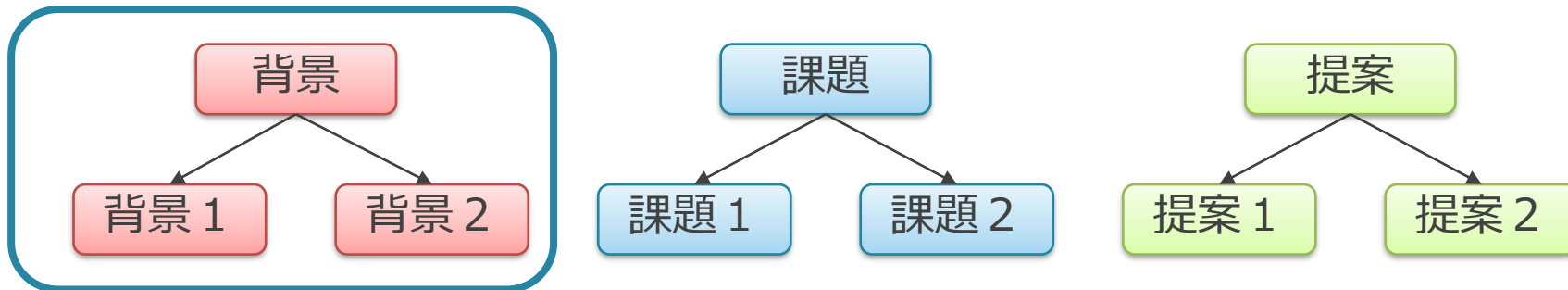
- フラットに並んでいる場合：全体を理解する=話題数 6



- 階層化されている場合：全体の話の大筋をつかむ=話題数 3



- 階層化されている場合：「背景」部分の詳細を理解する=話題数 3



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：

◇ 内容面：

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？

◇ 形式面：

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：

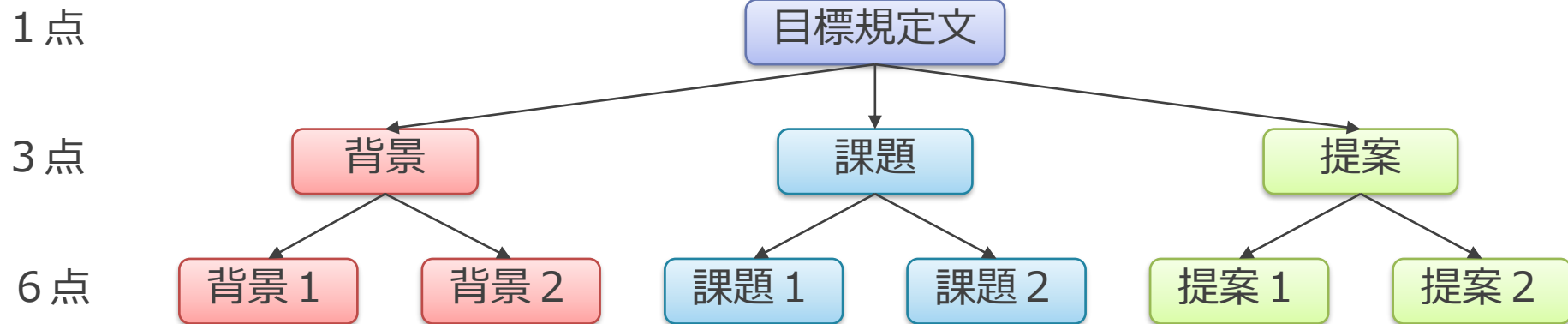
- ◇ 3点プロットが出来たら，そこからさらに真に重要な項目を抽出
- ◇ それらを繋げて目標規定文にする

6点プロット

6点プロット

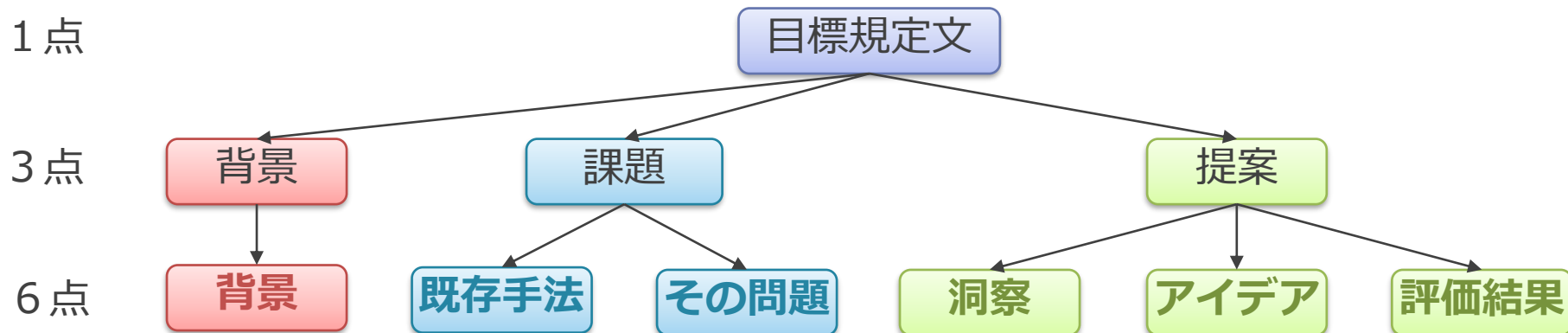
- 6点プロットはイントロを書く際に作る
 - ◇ 論文のイントロは典型的には6パラグラフ前後
 - ◇ この各パラグラフで何を話すかをまとめる
 - 6点プロットの各項目は各パラグラフのトピック・センテンスに対応する
 - トピック・センテンスだけを繋げて読んでも意味が通るように
- 6点プロットも3点プロットから派生させて作る
 - ◇ 3点プロットの各項目に、より詳細を肉付けしていく

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

6点プロット時の配分



■ 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、論文ごとに異なる

■ たとえば、

◇ 背景は1つのままで、提案を増やすことが多い（上の図の例）

◇ あまり一般的ではない話題の場合、背景が多めになる事も

◇ 課題の発見や整理こそが大事な場合、提案は自明なため短くなる

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、肉付けして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 subsection 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

スライド用プロットの流れの例

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. アイデア
 2. 実装：構成，動作，例
 3. 既存手法との比較
5. 評価
6. まとめ

◇ たとえば上記それぞれの項目に 1 ～ 4 ページ程度を割り当てる

プロットから文章へ

プロットから文章やスライドへ

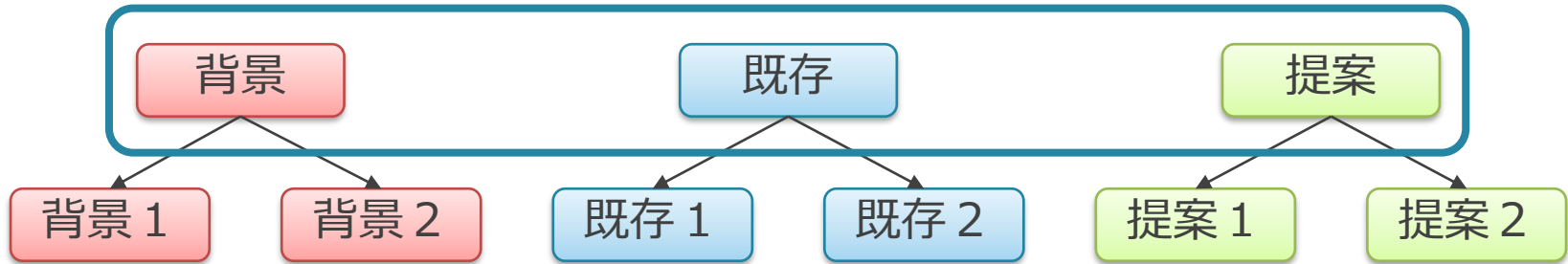
- プロットの文章の違い
 - ◇ プロットは論理の階層構造を持つ
 - ◇ しかし、文章（スライド）は基本的にシーケンシャル
 - 読者は前から後ろにむかって順に読む
- プロットの論理をどのようにシーケンシャルな文章に展開するか？

階層構造の展開の仕方

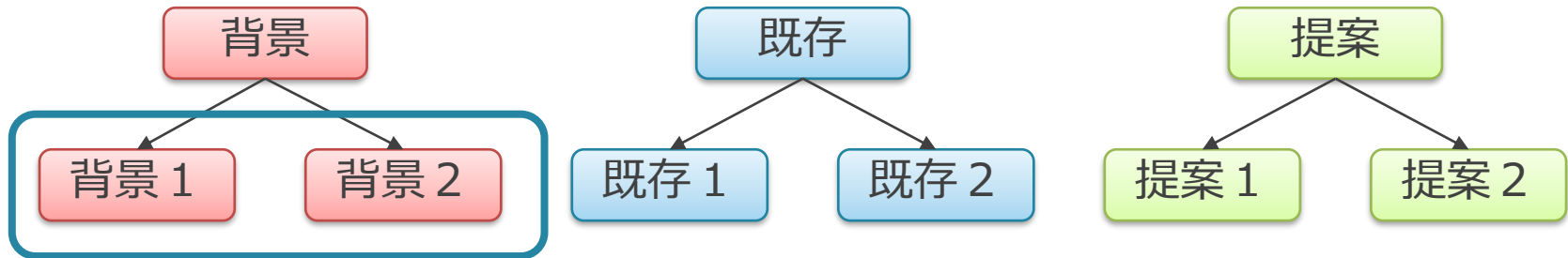
- 上から順に各階層にある話題を紹介したあと，1つずつ潜っていく
 - ◇ 典型的なやりかた：以下を再帰的に繰り返す
 - 登場人物（子）の紹介と，子同士の関係を説明
 - 各子の詳細を順に説明
- 典型例：
 - ◇ イン트로で論文全体の話題を紹介
 - ◇ 2節の冒頭で背景全体を簡単に説明
 - ◇ 2.1節で背景の1つめを説明
 - ◇ 2.2節で背景の2つめを説明
 - ◇ 3節の冒頭で背景との関係と共に既存手法全体を簡単に説明
 - ◇ 3.2節で既存手法の1つめを説明
 - ◇ 3.2節で既存手法の2つめを説明
 - ◇ ...

上から順に各階層にある話題を紹介したあと、 1つずつ潜っていく

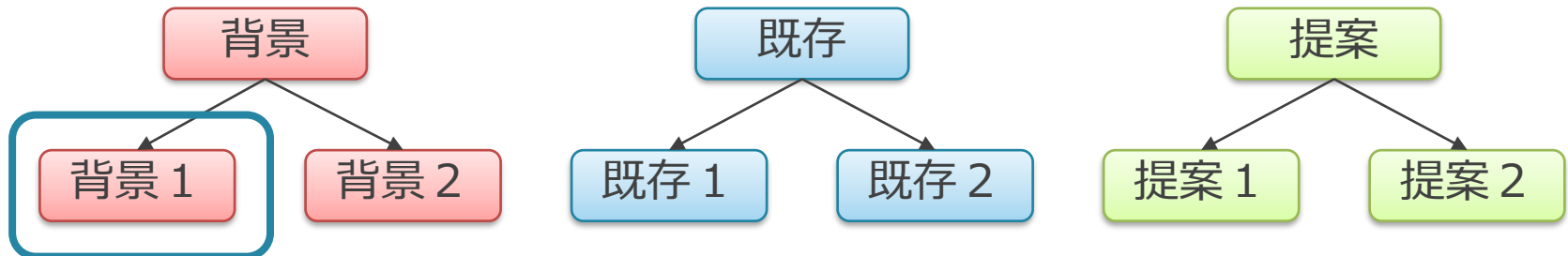
- イン트로で論文全体の話題（流れ）を紹介



- 2 節の冒頭で背景全体を簡単に説明

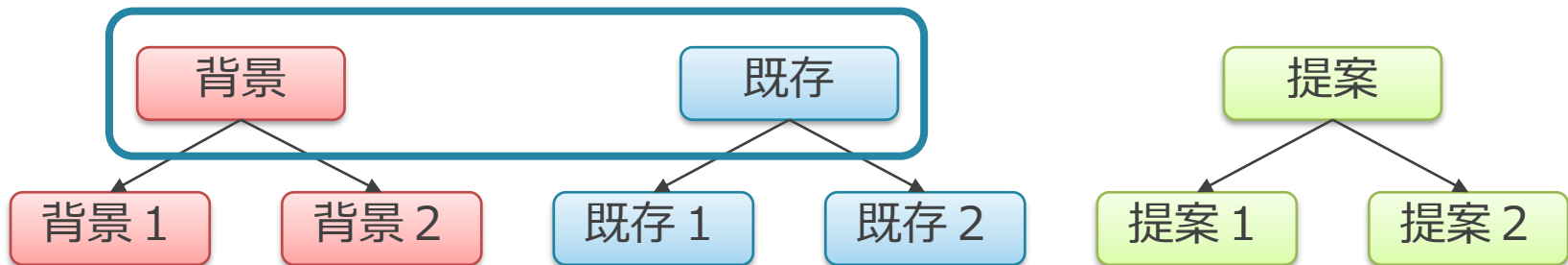


- 2.1 節で背景の 1 つめを紹介

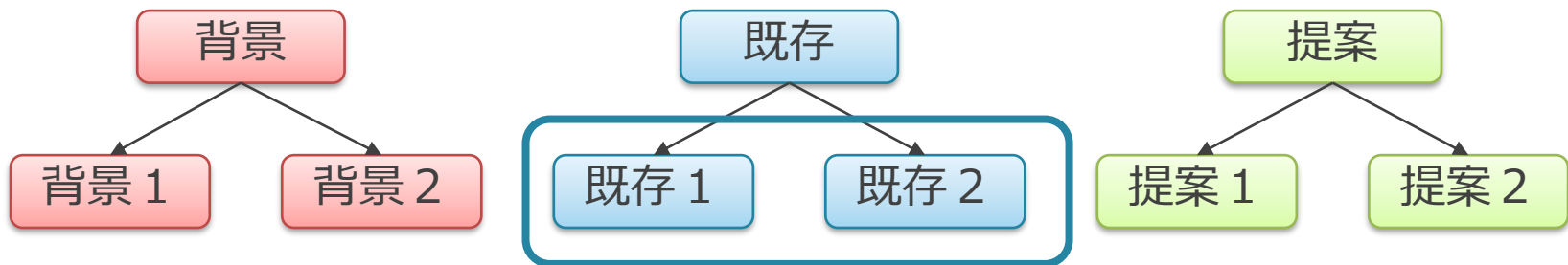


上から順に各階層にある話題を紹介したあと、 1つずつ潜っていく

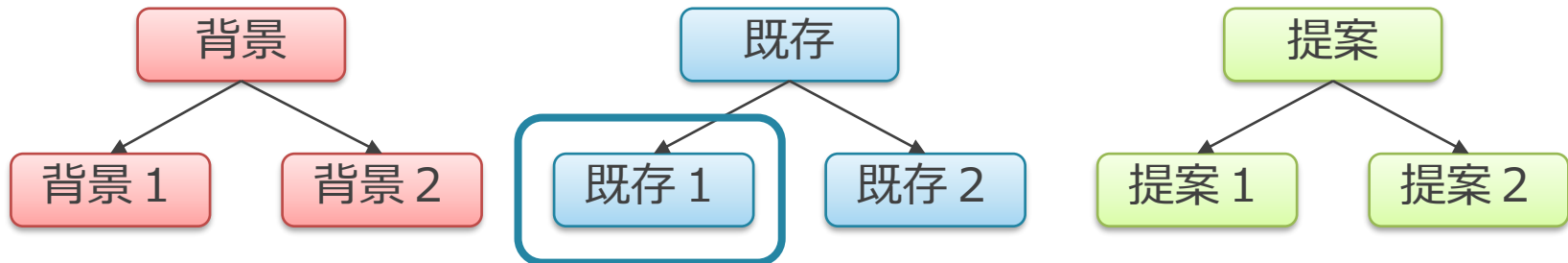
- 3 節の冒頭で背景との関係と共に



- 既存手法全体を簡単に説明



- 3.1 節で既存手法の 1 つめを説明



まとめ

まとめ

- プロットの作り方について説明
 - ◇ 1点から完全プロットまで
 - ◇ 3点プロットから始めるとよい
 - 背景, 課題, 提案の中身と関係をはっきりさせる
- 3点プロットは基本
 - ◇ 慣れてきたら6点プロット等から初めてもよい