

BDD ライブラリの紹介と SAPPOROBDD extended (構想)

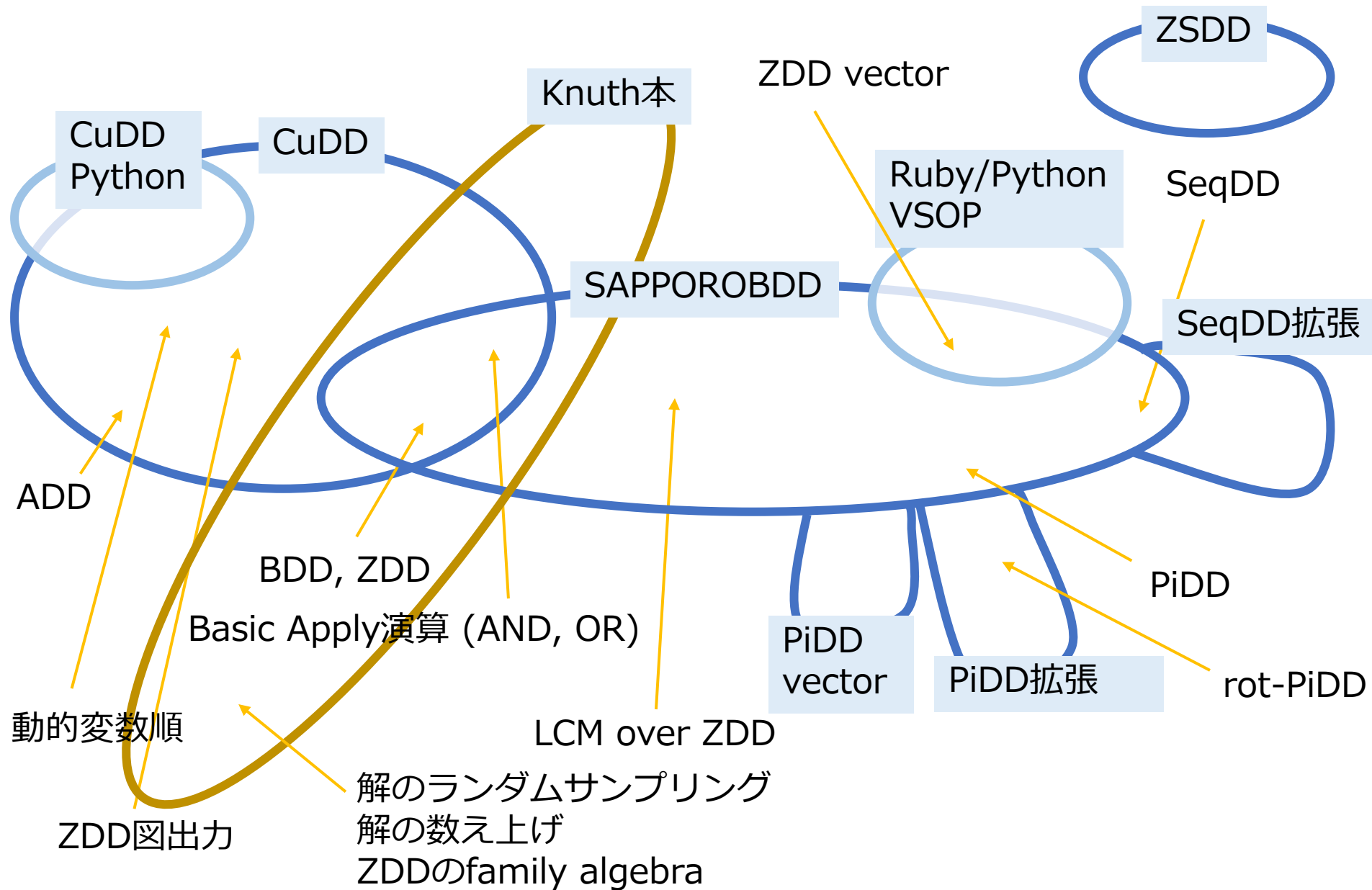
奈良先端科学技術大学院大学

川原 純

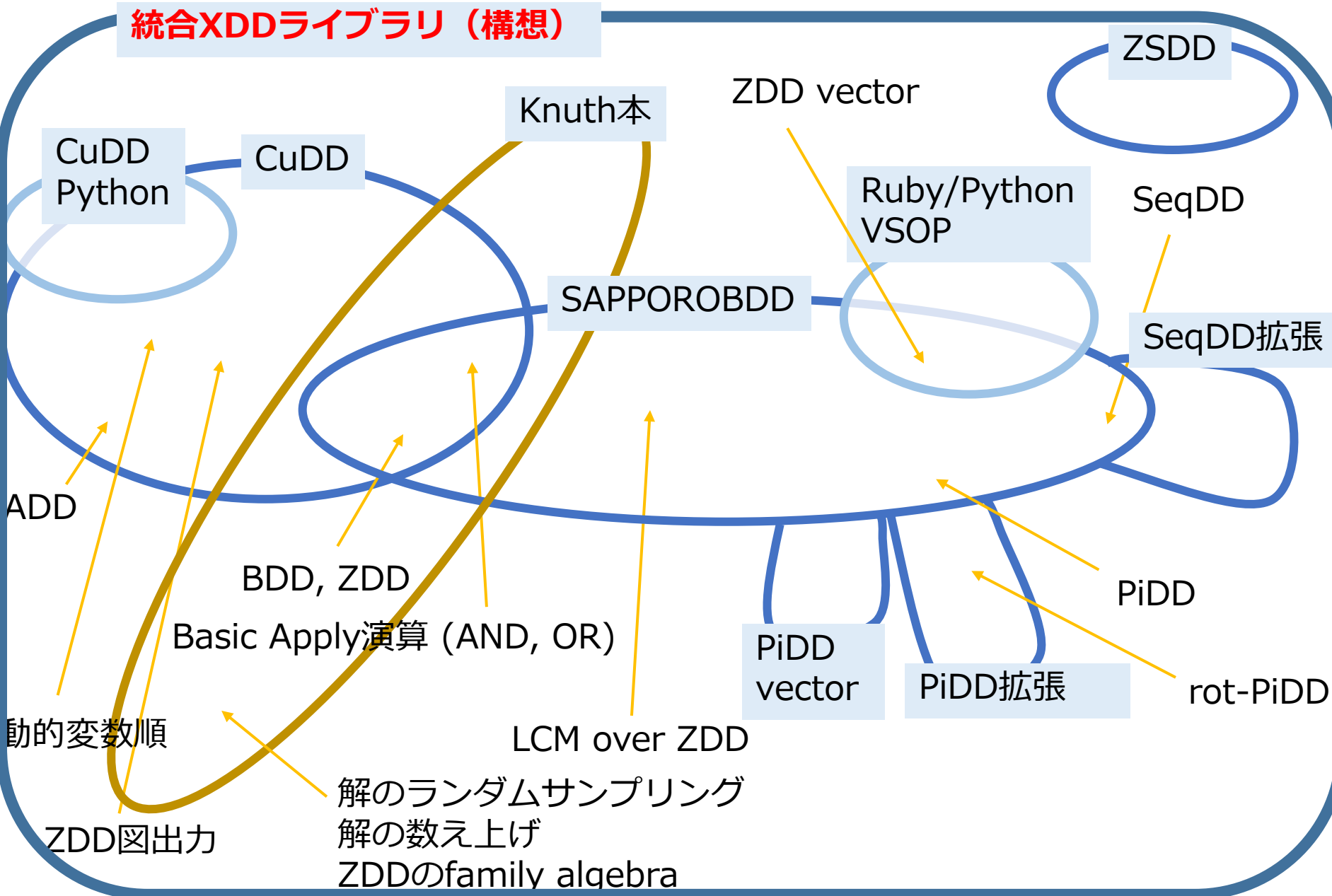
発表内容

- 既存のBDDライブラリを紹介
- SAPPOROBDDライブラリの拡張に向けて

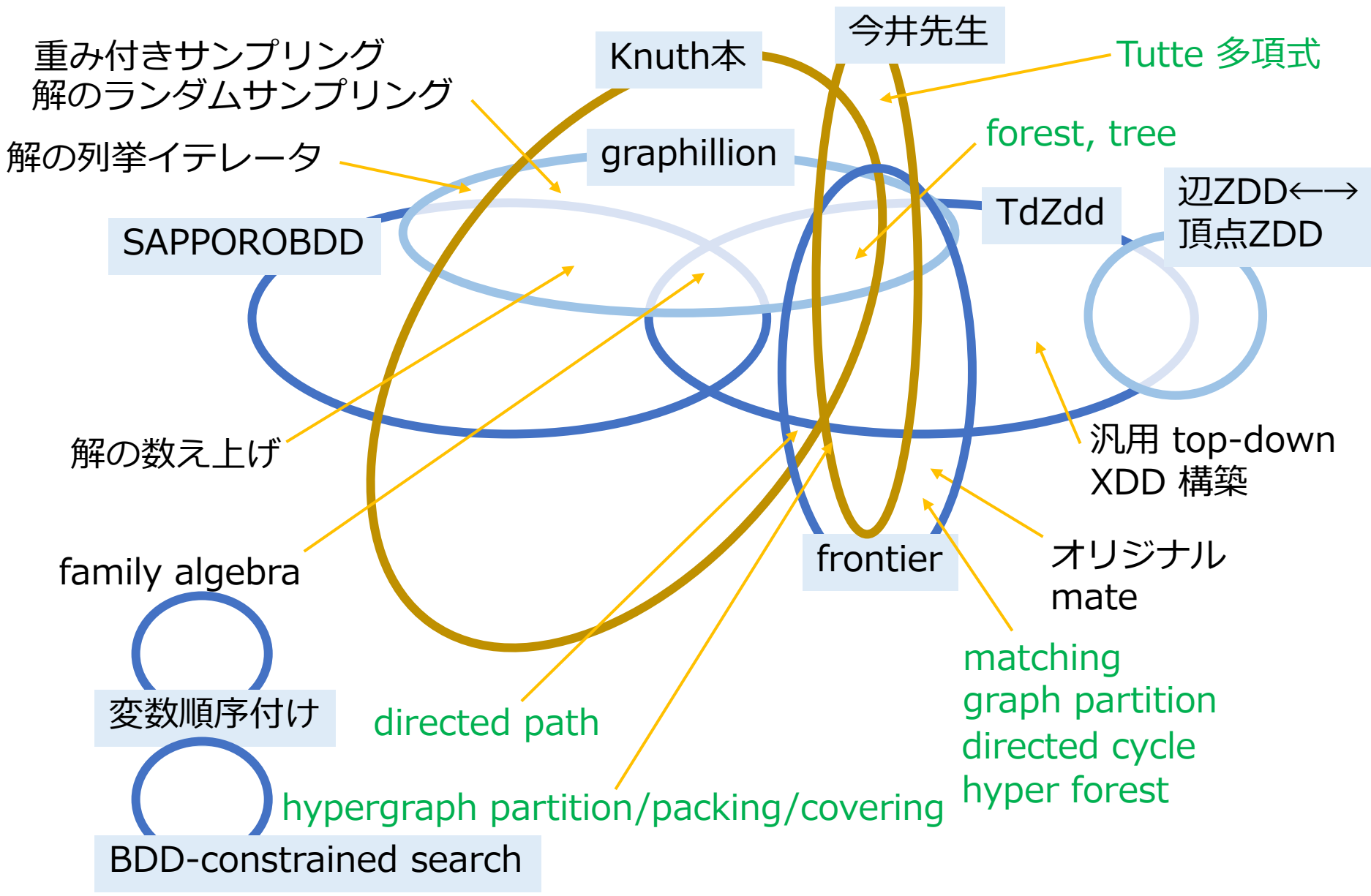
XDDライブラリの機能図



XDDライブラリの機能図

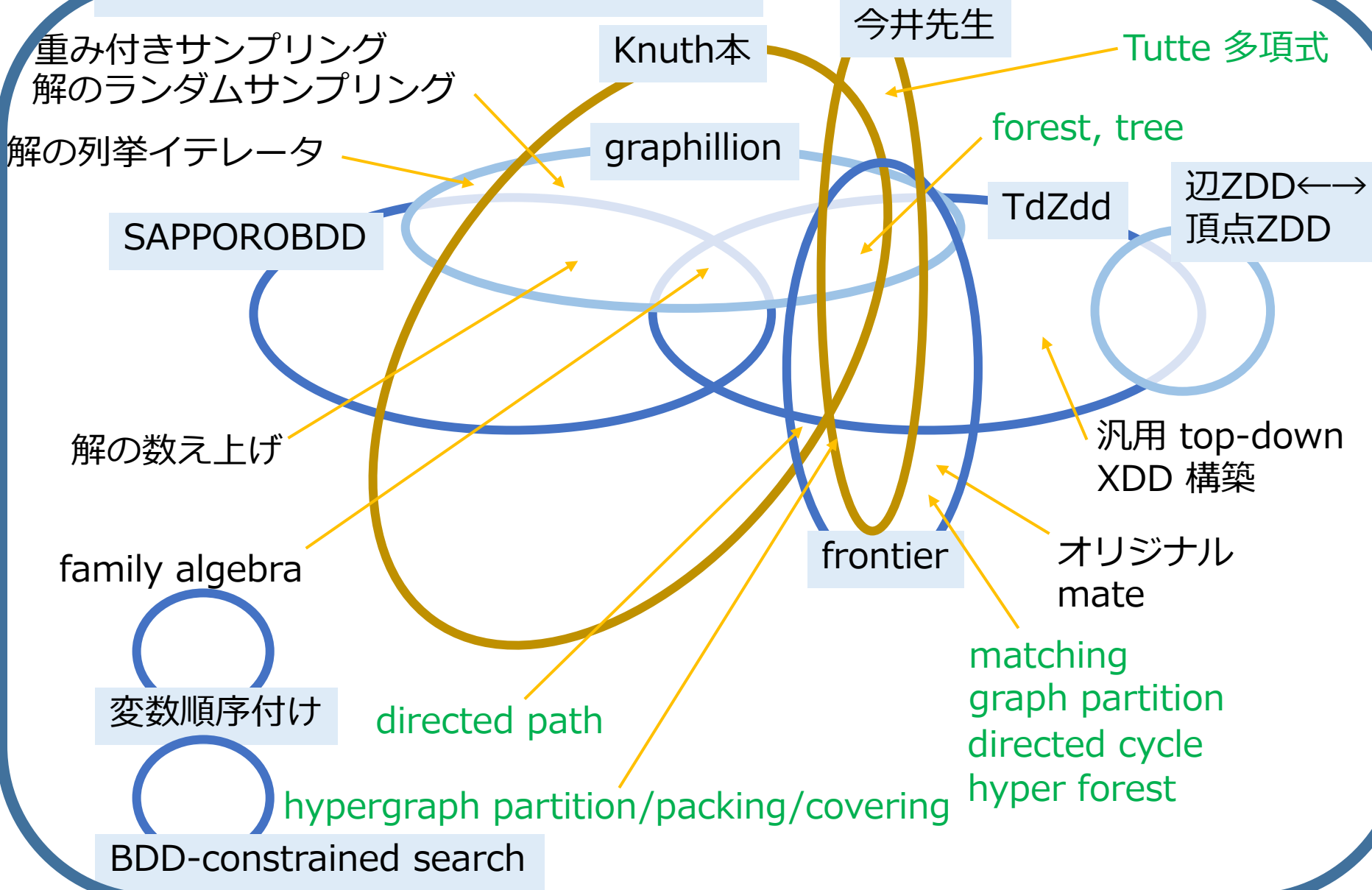


部分グラフ集合 (フロンティア法) ライブラリ機能図



部分グラフ集合 (フロンティア法) ライブラリ機能図

統合部分グラフ集合ライブラリ (構想)



CuDD

- Somenzi 先生 (Colorado 大) 作成
- BDD, ZDD, ADD ライブラリ
- C, C++ 言語
- 特徴
 - 世界中で最も使用されているBDDライブラリ
 - 信頼性が高い
 - 豊富な機能
 - 動的変数順、キャッシュとハッシュ、チューニングパラメタ、XDD図の出力
 - 唯一? のADDライブラリ (強化学習/POMDPで使用)
 - △ ドキュメントが分かりにくい
 - 解説が意外と少ない?
 - △ 湊ERATO/基盤(S) PJでの最新の研究が反映されていない
 - フロントティア法、SeqDD、PiDD などのサポートは一切ない

SAPPOROBDD

- 湊 真一先生（北海道大）作成
- BDD, ZDD, PiDD, SeqDD ライブラリ
- C, C++ 言語
- 特徴
 - 構造がシンプル（コアの bddc.c は3000行）
 - Apply 演算（XDDAnd など）が高速（CuDD と比べて2倍）
 - ZDD vector（多重集合）, PiDD, SeqDD を扱える唯一？のライブラリ
 - △ ユーザへの優しさは発展途上
 - △ XDDの内部構造を直接触ることが想定されていない
 - ✗ 正式公開されていない
 - 論文で引用できない

Ruby/Python VSOP

- 湊ERATO PJ 大阪グループ作
- 湊先生作の SAPPOROBDD や VSOP の Ruby/Python ラッパー
- 特徴
 - ユーザフレンドリ
 - XDDの様々な機能が利用できる
 - ただし、C++からそれらの機能呼び出すことはできない
 - △ XDDの内部構造を直接触ることが想定されていない
 - △ 名前がXDDライブラリっぽくない
 - 計算時間は未測定

graphillion

- 井上 武 氏作成 <http://graphillion.org>
- 部分グラフ集合ZDD（フロンティア法）ライブラリ。集合族も扱える
- C++, Python 言語
- 特徴
 - ユーザフレンドリ。XDDを知らなくても扱い可能
 - △ XDDの構造を触ることはできない
 - 機能が豊富
 - 一様ランダムサンプリング、重み付き最適化、イテレータ、subset、hitting set
 - △ 扱える部分グラフ構造は決められたものだけ
 - s-t パス、（根付き）（全域）森/木など
 - Python だから計算速度が遅い、ということはない
 - 論文として出版

TdZdd

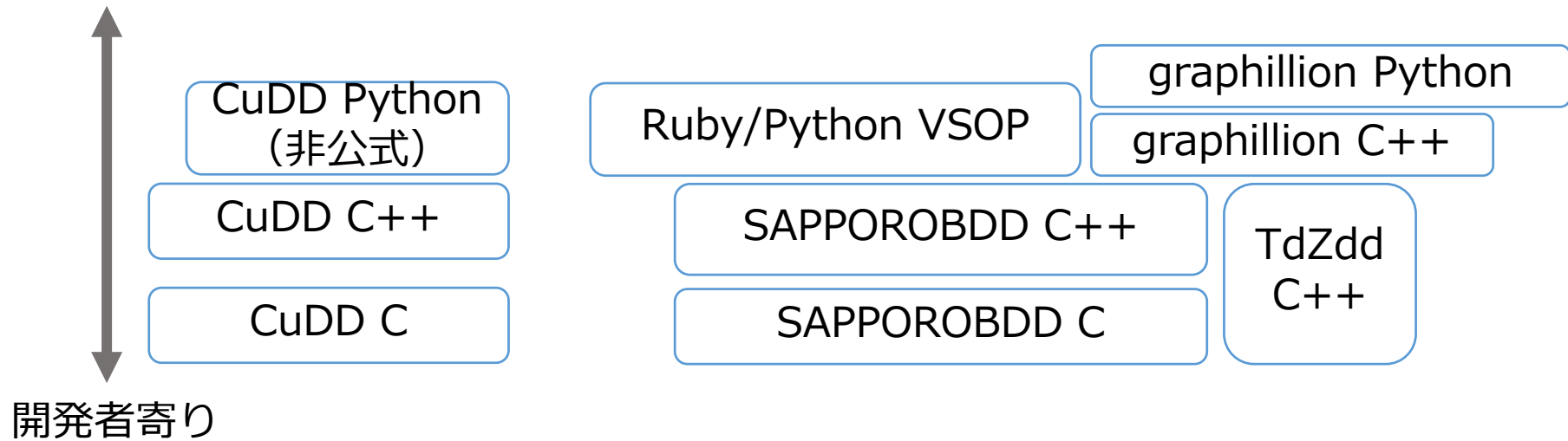
- 岩下 洋哲 氏作成
- トップダウンにXDDを作成するためのライブラリ（graphillion の内部でも使用）
- C++ 言語
- 特徴
 - 汎用的
 - △ 汎用的なため、使いこなすためにはXDDを熟知している必要がある
 - △ フロントティアは自分で実装する必要がある
 - 高速
 - △ ドキュメントが少ない。やりたいことを実現する方法が分かりにくい
 - 最近、TdZddの使い方の解説論文が出版された

戸田 貴久, 斎藤 寿樹, 岩下 洋哲, 川原 純, 湊 真一,
"ZDDと列挙問題 - 最新の技法とプログラミングツール,"
コンピュータソフトウェア, 2016. (to appear)

XDDライブラリのレイヤー図

• レイヤー図

ユーザ寄り



frontier プログラム

- 川原（発表者）作成
- フロンティア法で様々な部分グラフ構造を表す XDDを構築するプログラム
- C++ 言語
- 特徴
 - フロンティア法で、XDDノードに蓄える情報（通称 mate）を自由に指定してXDDを構築できる
 - 現状、唯一？有向グラフやハイパーグラフを扱える
 - △ ライブラリではない（ただのプログラム）
 - フロンティア法のノード作成を、「Mate クラス」を継承して記述
 - △ フロンティア法について詳しいことが前提
 - △ XDDを構築した後は、別のライブラリに任せる
 - ✗ ドキュメントがない

SeqDD プログラム

- 伝住 周平先生（東京大）作成
- SeqDD（文字列の集合）処理系
- C++/erlang で作成？
- 特徴
 - ✕ 非公開
 - 伝住先生の手元にのみ存在
 - C++ 実装はSAPPOROBDDの上で拡張する形
 - SAPPOROBDDの機能がそのまま使える
 - 豊富な演算体系

SeqDD プログラム

• 伝住 周平先生 (東京大) 作成

表 2 SeqBDD の発展演算

演算	返す文字列集合
Concat(P, Q)	$\{xy x \in P, y \in Q\}$.
Weld $_{\alpha\beta\gamma}(P, Q)$	$\{xyz xy \in P, yz \in Q,$ $ x \geq \alpha, y \geq \beta, z \geq \gamma\}$.
LDiv(P, Q)	$\{z y \in Q, yz \in P\}$.
RDiv(P, Q)	$\{x y \in Q, xy \in P\}$.
CDiv(P, Q)	$\{xz y \in Q, xyz \in P\}$.
LRem(P, Q)	$\{w w \in P, \exists y \in Q, w \neq yz, \forall z \in \Sigma^*\}$.
RRem(P, Q)	$\{w w \in P, \exists y \in Q, w \neq xy, \forall x \in \Sigma^*\}$.
CRem(P, Q)	$\{w w \in P, \exists y \in Q, w \neq xyz, \forall x, z \in \Sigma^*\}$.
LDel(P, Q)	$\{z \exists y \in Q, yz \in P\}$.
RDel(P, Q)	$\{x \exists y \in Q, xy \in P\}$.
CDel(P, Q)	$\{x_0 \cdots x_n \exists y_1, \dots, y_n \in Q,$ $x_0y_1x_1 \cdots y_nx_n \in P,$ $Q \cap \text{Factor}(x_i) = \emptyset, i = 0, \dots, n\}$.
LDelRem(P, Q)	$\{w w \in P, y \in Q, w \neq yz, \forall z \in \Sigma^*\}$.
RDelRem(P, Q)	$\{w w \in P, y \in Q, w \neq xy, \forall x \in \Sigma^*\}$.
CDelRem(P, Q)	$\{w w \in P, y \in Q, y \notin \text{Factor}(w)\}$.
PrefReplace(P, Q, S)	$\{wz \exists y \in Q, yz \in P, w \in L(S),$ or $y \in Q, wz \in P, y \notin \text{Prefix}(wz)\}$.
SuffReplace(P, Q, S)	$\{xw \exists y \in Q, xy \in P, w \in L(S),$ or $y \in Q, xw \in P, y \notin \text{Suffix}(xw)\}$.
FactReplace(P, Q, S)	$\{x_0w_1 \cdots w_n \exists y_1, \dots, y_n \in Q,$ $x_0y_1x_1 \cdots y_nx_n \in P, Q \cap \text{Factor}(x_i) = \emptyset,$ $i = 0, \dots, n, w \in L(S)$ or $y \in Q, xwz \in P, y \notin \text{Factor}(xwz)\}$.
Insert(P, Q)	$\{xyz zx \in P, y \in Q\}$.
Separate(P, Q)	$\{x_i \exists y_1, \dots, y_n \in Q, x_0y_1x_1 \cdots y_nx_n \in P,$ $Q \cap \text{Factor}(x_i) = \emptyset, i = 0, \dots, n$ $x_0 \in P, y \in Q, y \notin \text{Factor}(x_0), i = 0\}$.
Rev(P)	$\{x^R x \in P\}$.
LRotate(P)	$\{xa ax \in P, a \in \Sigma\} \cup (P \cap \{\epsilon\})$.
RRotate(P)	$\{ax xa \in P, a \in \Sigma\} \cup (P \cap \{\epsilon\})$.

Rev に関しては、青木らによって効率の良いアルゴリズムが提案されている [3]。これらのうち、Concat, Weld(α, β, γ は 0 も

表 3 SeqBDD の索引演算

演算	返す文字列集合
PSDD(P)	$\{x w \in P, x \in \text{Prefix}(w)\}$.
SSDD(P)	$\{x w \in P, x \in \text{Suffix}(w)\}$.
FSDD(P)	$\{x w \in P, x \in \text{Factor}(w)\}$.
SubSDD(P)	$\{x w \in P, x \in \text{Subseq}(w)\}$.
propPSDD(P)	$\{x w \in P, x \in \text{Prefix}(w), x \neq w\}$.
propSSDD(P)	$\{x w \in P, x \in \text{Suffix}(w), x \neq w\}$.
propFSDD(P)	$\{x w \in P, x \in \text{Factor}(w), x \neq w\}$.
propSubSDD(P)	$\{x w \in P, x \in \text{Subseq}(w), x \neq w\}$.
HammingDist(P, d)	P に含まれる文字列からハミング距離 d 以内の文字列からなる集合。
LevenDist(P, d)	P に含まれる文字列からレーベンシュタイン距離 d 以内の文字列からなる集合。
Dam - LevenDist(P, d)	P に含まれる文字列からタメロウ・レーベンシュタイン距離 d 以内の文字列からなる集合。
Anagram(P)	P に含まれる文字列のアナグラムからなる集合

表 4 SeqBDD の抽出演算

演算	返す文字列集合
Shorter(P, l)	$\{x x \in P, x \leq l\}$.
Longer(P, l)	$\{x x \in P, l \leq x \}$.
Just(P, l)	$\{x x \in P, l = x \}$.
Shortest(P)	$\{x x \in P, x = \min_{y \in P} \{ y \}\}$.
Longest(P)	$\{x x \in P, x = \max_{y \in P} \{ y \}\}$.
ExistPref(P, Q)	$\{x x \in P, \exists y \in Q, y \in \text{Prefix}(x)\}$.
ExistSuff(P, Q)	$\{x x \in P, \exists y \in Q, y \in \text{Suffix}(x)\}$.
ExistFact(P, Q)	$\{x x \in P, \exists y \in Q, y \in \text{Factor}(x)\}$.
ExistSubseq(P, Q)	$\{x x \in P, \exists y \in Q, y \in \text{Subseq}(x)\}$.
PrefMaximal(P)	$\{x x, y \in P, x \neq y, x \notin \text{Prefix}(y)\}$.
SuffMaximal(P)	$\{x x, y \in P, x \neq y, x \notin \text{Suffix}(y)\}$.
FactMaximal(P)	$\{x x, y \in P, x \neq y, x \notin \text{Factor}(y)\}$.
SubseqMaximal(P)	$\{x x, y \in P, x \neq y, x \notin \text{Subseq}(y)\}$.
PrefMinimal(P)	$\{x x, y \in P, x \neq y, y \notin \text{Prefix}(x)\}$.
SuffMinimal(P)	$\{x x, y \in P, x \neq y, y \notin \text{Suffix}(x)\}$.
FactMinimal(P)	$\{x x, y \in P, x \neq y, y \notin \text{Factor}(x)\}$.
SubseqMinimal(P)	$\{x x, y \in P, x \neq y, y \notin \text{Subseq}(x)\}$.

伝住 周平, 有村 博紀, 湊 真一, 系列二分決定グラフを操作するための豊富な演算体系の構築, コンピューテーション研究会, 2012

PiDDプログラム

- 井上 祐馬 氏（北海道大）作成
- PiDD を扱うプログラム
- ???言語
- 特徴
 - ✕非公開
 - C++ 実装はSAPPOROBDDの上で拡張する形???
 - SAPPOROBDDの機能がそのまま使える
 - 豊富な演算体系
 - rot-PiDD など、PiDDの別の表現も扱える

その他のXDD関連プログラム

- 頻出パターン列挙: LCM over ZDD (SAPPOROBDD付属)
 - 宇野 毅明 先生 (NII)、湊先生
- 部分グラフ集合で、辺変数ZDDと頂点変数ZDDを相互変換するプログラム (単独公開)
 - 鈴木 浩史 氏 (北海道大)
- tree-decomposition に基づく辺変数順序付け (graphillion付属)
 - 井上 祐馬 氏
- BDD-constrained Search, A*探索 (非公開)
 - 西野 正彬 氏、安田 宜仁 氏、竹内 文登 氏
- 極小ヒッティングセットの計算とSATの全解列挙 (単独公開)
 - 戸田 貴久 先生 (電通大)
- PiDD vector (非公開)
 - 川原
- ZSDD (Zero-Suppressed Sentential Decision Diagrams) (非公開)
 - 西野 正彬 氏
- おねえさん問題 (単独公開)
 - 岩下 洋哲 氏
- ...

プログラムパッケージの開発モデル例

- SageMath

- 世の中にある、様々なオープンソースの数学プログラム・ライブラリを統合したパッケージ
- Python言語による共通インターフェイス 引用: wikipedia (Sage)

計算機代数	GAP, Maxima, SINGULAR
多倍長計算	MPIR, MPFR, MPFI, NTL
数式処理	Maxima, SymPy, GiNac
組合せ	Symmetrica, Sage-Combinat
線形代数	ATLAS, BLAS, LAPACK, MumPy, GSL
グラフ理論	NetworkX
群論	GAP
数値計算	GSL, SciPy, NumPy, ATLAS
統計処理	R, SciPy
その他	IPython, SQLite, jsmath, Matplotlib

- XDD にも、SageMath のような統合パッケージが欲しい
 - 将来的には XDD パッケージも SageMath に入れば嬉しい

SAPPROBDDのAPI

```
// 集合族 {{5}} を表すZDD
```

```
ZDD z;
```

```
z = ZBDD(1);
```

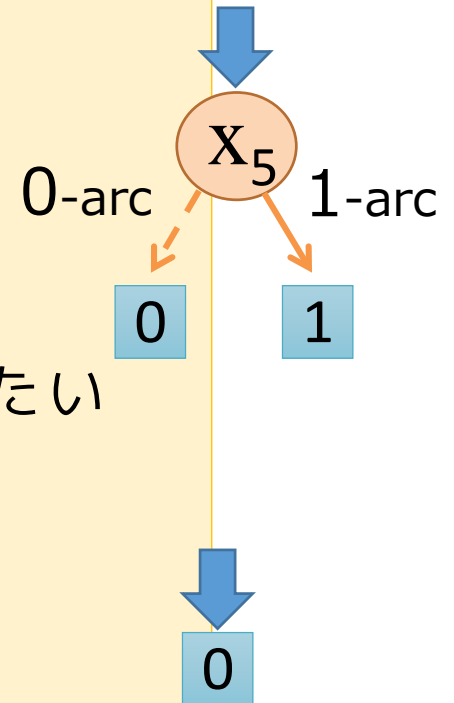
```
z = z.Change(5);
```

```
// z = ZDD::GetSingleton(5); と書きたい
```

```
// z が {} かどうか判定
```

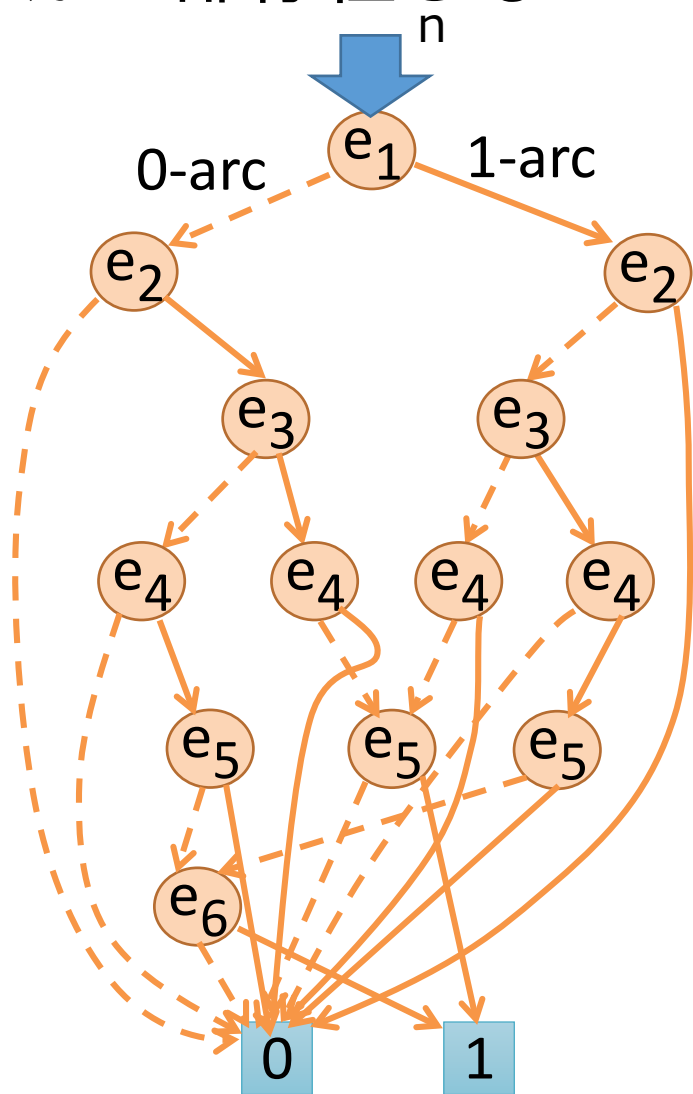
```
if (z == ZBDD(0)) ...
```

```
// if (z.IsZeroTerminal()) ... と書きたい
```

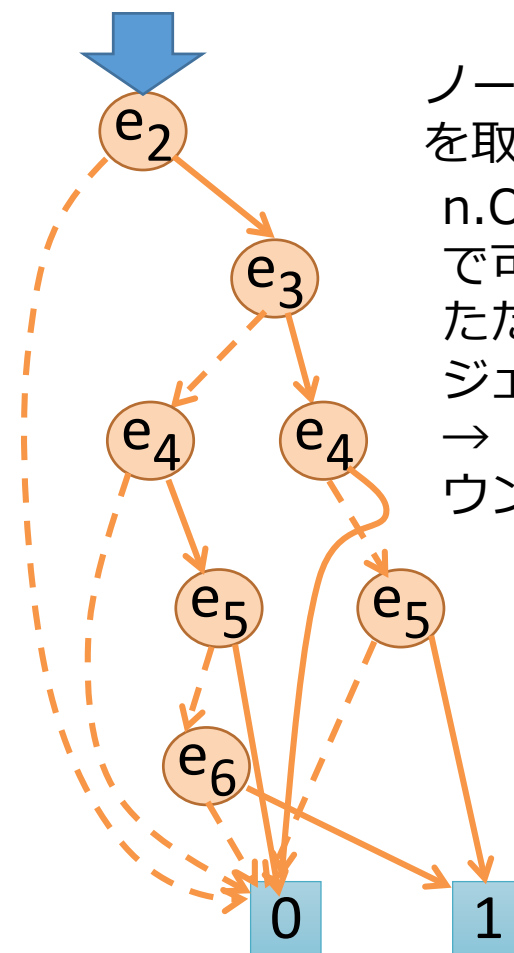


BDDの0-arc取得

- SAPPOROBDD には内部構造を操作する機能が一部存在しない



ノード n の 0-arc 側を取得



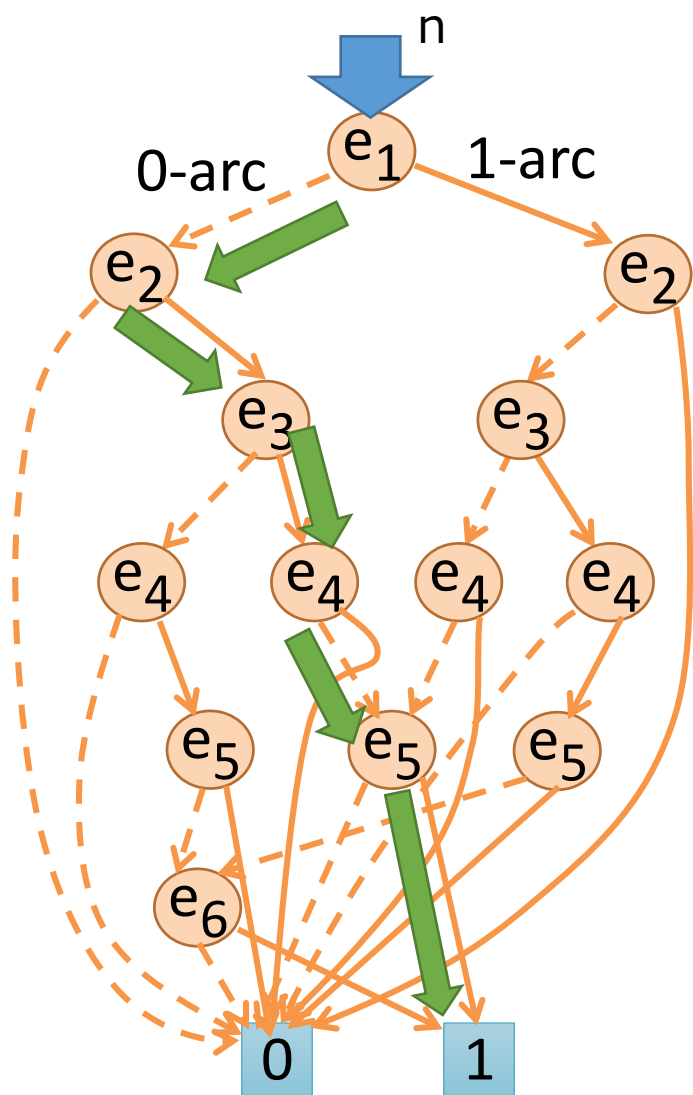
ノード n の 0-arc 側を取得

`n.Offset(n.Top())` で可能。
ただし、ZDDオブジェクトが作られる
→ リファレンスカウンタを増やす等

実際は否定枝を使用して実装されているので、単純ではない

0/1-child 取得

- 予備実験



ZDDの根から、確率1/2で0-arcを、確率1/2で1-arcを選択して下りる。終点にたどり着いたら根に戻る。arcの選択を1,000,000 回行う。

CPU Xeon E5-2630 2.3GHz /
memory 128GB / Linux Centos
6.6 / C++ / g++ v4 -O3

プログラム	計算時間 (sec.)
現 C++ API	11.12
現 C API	8.93
getchild	4.10
getchild inline	4.04
getchild inline NDEBUG	3.37

C++ API: `n.Offset(n.Top())`

C API: `bddoffset(n, bddtop(n))`

getchild: 内部構造から直接取得

SAPPOROBDD extended 構想

- XDDライブラリ統合のために、まずは SAPPOROBDD を強化する
- SAPPOROBDD のソースコード bddc.c、bddc.h、ZBDD.cc など是一切触らずに、関数やクラスを追加する形で拡張する。
 - SAPPOROBDD のソースコード自体は graphillion の一部として公開されている（正式公開ではない）
 - 湊先生のご負担を避ける

bddc_ex.c

```
#include "bddc.c"

bddp getchild(bddp p, int c) {
    ...
}
```

- 基本的には、川原が過去に書いたコード片をベースにする

SAPPOROBDD extended 設計思想

- XDD や部分グラフ集合ZDD のことのある程度以上分かっている人向け
 - 言語は C/C++
 - Python のラッパーがあってもよい (将来)
- XDD のアルゴリズムの研究をしている際によく書くことになるユーティリティを提供
- XDDの内部構造をいじることが可能

既に実装した機能

- getchild

```
inline bddp bddgetchild0b_nc(bddp f)
inline bddp bddgetchild0z_nc(bddp f)
inline bddp bddgetchild0_nc(bddp f)
inline bddp bddgetchild0braw_nc(bddp f)
inline bddp bddgetchild0zraw_nc(bddp f)
inline bddp bddgetchild0raw_nc(bddp f)
inline bddp bddgetchild1b_nc(bddp f)
inline bddp bddgetchild1z_nc(bddp f)
inline bddp bddgetchild1_nc(bddp f)
inline bddp bddgetchild1braw_nc(bddp f)
inline bddp bddgetchild1zraw_nc(bddp f)
inline bddp bddgetchild1raw_nc(bddp f)
inline bddp bddgetchildb_nc(bddp f, int child)
inline bddp bddgetchildz_nc(bddp f, int child)
inline bddp bddgetchild_nc(bddp f, int child)
inline bddp bddgetchildbraw_nc(bddp f, int child)
inline bddp bddgetchildzraw_nc(bddp f, int child)
inline bddp bddgetchildraw_nc(bddp f, int child)
```


既に実装した機能

- ZBDD_CountSolutions
 - ZDDに含まれる集合の個数を、任意桁整数で返す
- ZBDD_SampleRandomly
 - メルセンヌツイスターを乱数生成器として、ZDDから集合を一様ランダムサンプリング
 - ex.) 18×18グリッド
17821128408420651298933849466523252751678
38065704767655931452474605826692782532 本
のパスから、一様ランダムに1本サンプリングできる
- ZBDD_GetIndex
 - 引数で与えた集合が、ZDD（集合族）において辞書順で何番目かを、任意桁整数で返す
- ZBDD_GetIthSet
 - ZDD（集合族）に含まれる集合のうち、辞書順で引数で与えた任意桁整数 番目の集合を返す

既に実装した機能

- ZBDD_Output
 - ZDDを graphillion 形式でファイル/標準出力
- ZBDD_Visit
 - ZDDのノードを訪れる（コールバックが呼び出される）
- ZBDD_VisitInLevelOrder
 - ZDDのノードをレベル順を訪れる
- ZBDD_PrintElements
 - ZDD内の各要素をファイル/標準出力
- ZBDD_VisitToElements
 - ZDD内の各要素を訪れる（要素ごとにコールバックが呼び出される）

まとめ

- SAPPOROBDD extended は公開予定
- 自分の書いたコードを可能な限り github で公開すると、喜ぶ人が多い
- 将来的にXDD統合パッケージの開発に加わっていただけると嬉しいです