

Formulation modale

Une représentation innovante
de la connaissance pour réaliser
les systèmes d'information du futur

Pierre Gradiat, Michel Vandenberghe
SARL MEZZONOMY – 2016Q2

Systemes Décentralisés Interactifs

- Où toute action sur le SDI est un acte à valeur juridique
→ daté et certifié conforme.
- Où tout sujet (acteur ou association) est propriétaire de ses actes.
→ possibilité de réaliser des transactions entres pairs ou collectives.
- Où chaque acteur peut naviguer à volonté dans toutes les dimensions structurantes du SDI
→ navigation « *régulière* » et interactive.
- Où le nombre de dimensions structurantes n'est pas fixé a priori
→ dimensions de socialisation ou « *dimensions métier* »
- Où le temps est une dimension structurante
→ Tout objet est une vue sur son processus d'élaboration.
- Où les normes du W3C sont applicables
→ XML, DOM, XSL, ...
- Où le nombre de serveurs et d'acteurs n'est pas borné a priori
→ Créer des passerelles avec tout autre système, pair ou tiers.

Nécessité d'une structure innovante

- Les **graphes** sont à la base de toute représentation de la donnée socialisée
 - Problème : les graphes ne sont pas des supports robustes pour des calculs
 - Cas d'usage : calcul du graphe dual
- Nécessité d'une forme calculable des graphes
 - Capable de traiter les données semi-structurées
 - Capable de représenter les graphes de façon efficiente
 - Capable de intégrer le temps dans la représentation

Plan de l'exposé

- Créer une topologie calculable
- Traiter les données semi-structurées
- Intégration du temps
- Conclusion & Perspectives

"Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius - and a lot of courage - to move in the opposite direction. "

Albert Einstein

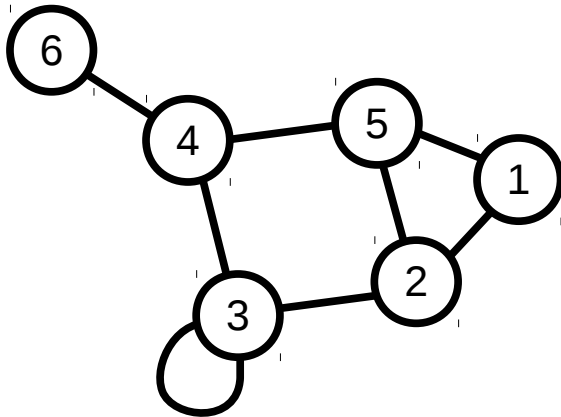
Plan de l'exposé

- **Créer une topologie calculable**
 - Théorie « *classique* » des graphes
 - Théorie « *modale* » des graphes
 - Dualité « *classique* » et « *modale* »
- Traiter les données semi-structurées
- Intégration du temps
- Conclusion & Perspectives

Créer une topologie calculable

- **Le graphe sont à la base de toute représentation du complexe :**
 - les documents « *complexes* » de définition du W3C font un usage courant de la forme graphique pour aider la lecture.
- La théorie des graphes souffre de plusieurs faiblesses structurelles :
 - **Pas de forme normale des graphes** : chacun à la sienne (ensembliste, fonctionnelle, matricielle, ...) et aucune ne s'est imposée. La page wikipédia sur la théorie des graphes compte sept représentations différentes.
 - **Relation ambiguë avec la topologie du plan** : tout graphe est lié à un dessin mais les problèmes du dessin, le plongement du graphe dans la topologie du plan, sont souvent éludés car assez friables.
 - **Faible accointance avec la notion de calcul** : nous pouvons constater que les preuves sur les graphes sont plus intuitives (en langage naturel) que calculables (en langage artificiel).

Théorie « *classique* » des graphes



Parmi la multitudes des écritures imaginables nous suivrons l'idée de Berge (1960) :

- $\{\}$: Ensemble
- $[0-9]^*$: Nombre servant de « *nom* »

Le « *graphe wikipédia* »

https://fr.wikipedia.org/wiki/Théorie_des_graphes

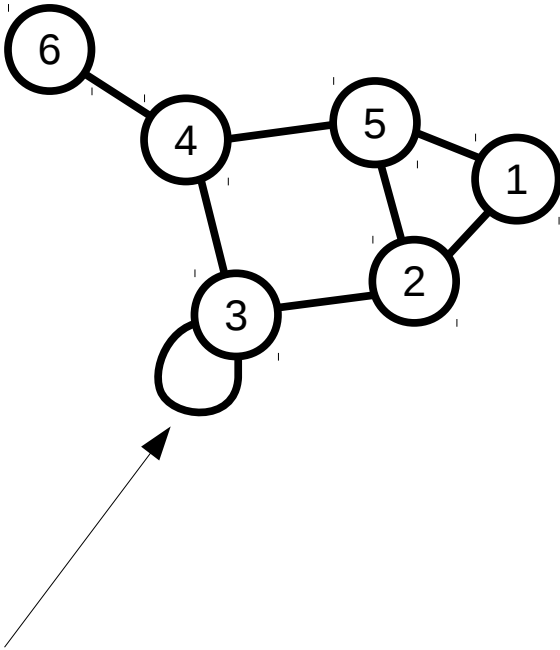
Le « *graphe wikipédia* » s'écrirait alors :

(
V : {1, 2, 3, 4, 5, 6, },
E : {{1,2}, {2,3}, {2,5}, {3,3}, {4,3}, {4,5}, {4,6},
)

↖ Zut, c'est pas un ensemble !
Chut ! C'est pas grave

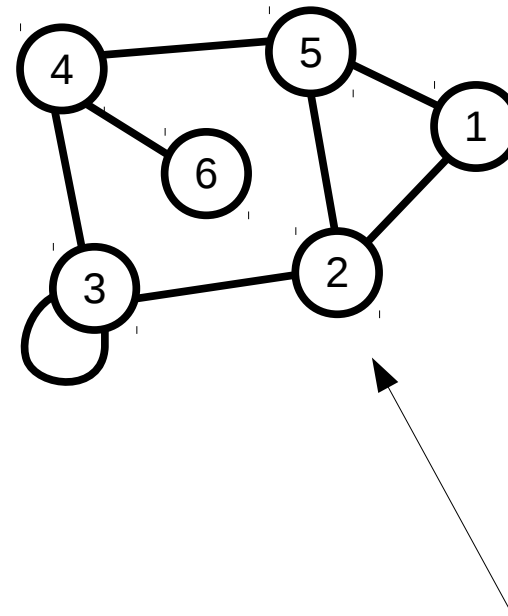
La question du dessin

i.e. du « *plongement dans la topologie du plan* »...



Le « *graphe wikipédia* »

https://fr.wikipedia.org/wiki/Théorie_des_graphes

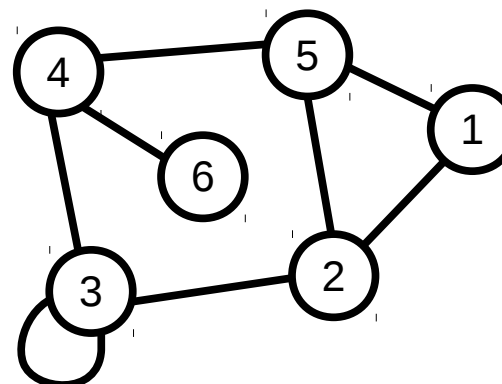
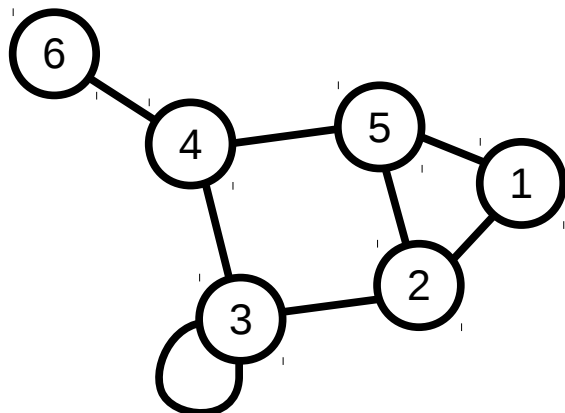


Le « *même* » avec le « 6 » à « *l'intérieur* »

Est-ce le « *même* » ?

Problématique clé de la calculabilité (Birkoff – 1933)

Oui, ce sont les « *mêmes* »...



|| EGALITE SYNTAXIQUE ||

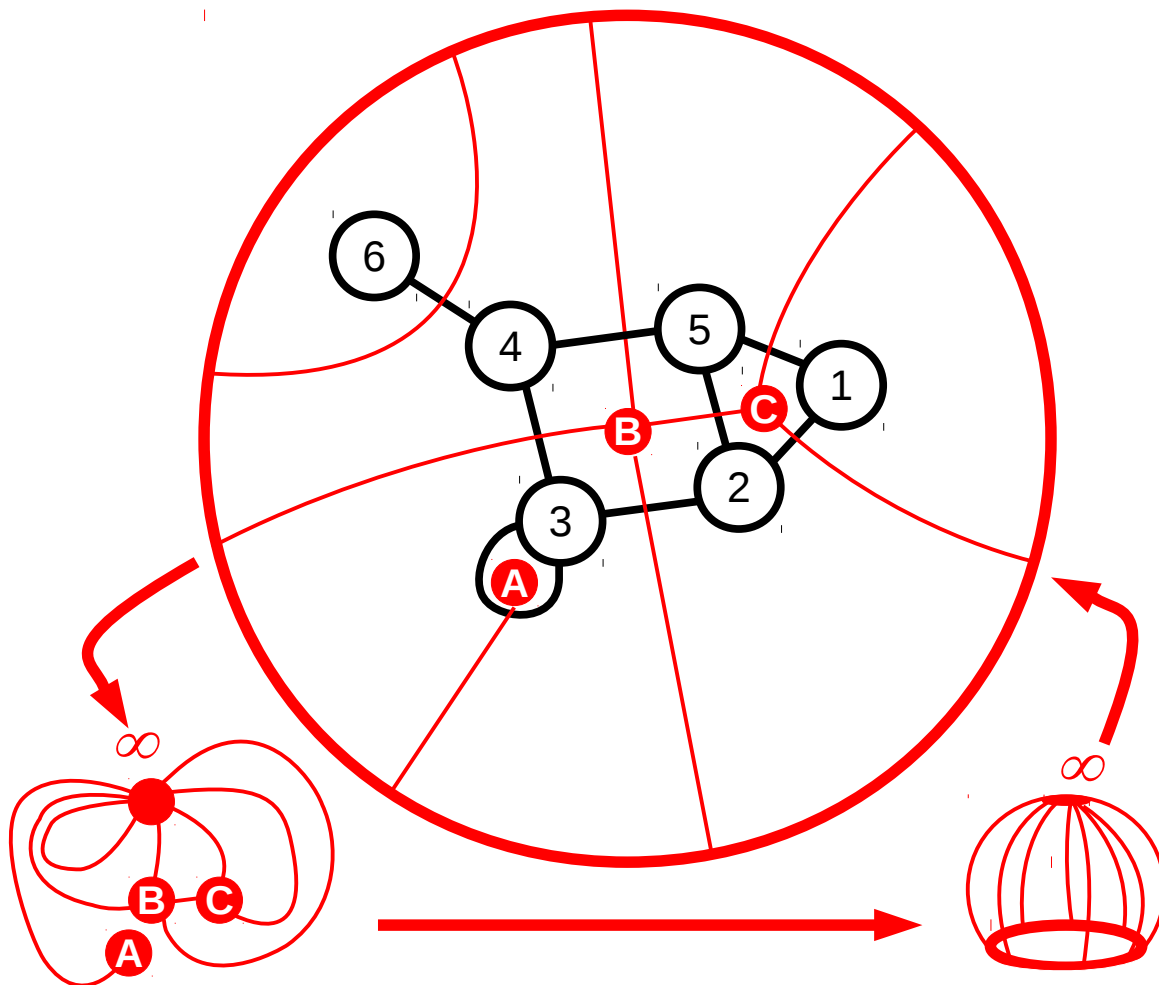
```
{  
  V : {1, 2, 3, 4, 5, 6, },  
  E : {{1,2}, {2,3}, {2,5}, {3,3}, {4,3}, {4,5}, {4,6},  
}
```

*Pourtant, je vois pas la même chose
Chut, te dis-je ! C'est pas grave !*

*Ont-ils le même graphe dual ?
Si le dual est un calcul, oui.*

Dualité « classique » des graphes

Le dual se construit sur l'ensemble des « faces » et en interprétant chaque arête comme une « frontière » entre deux « faces ».



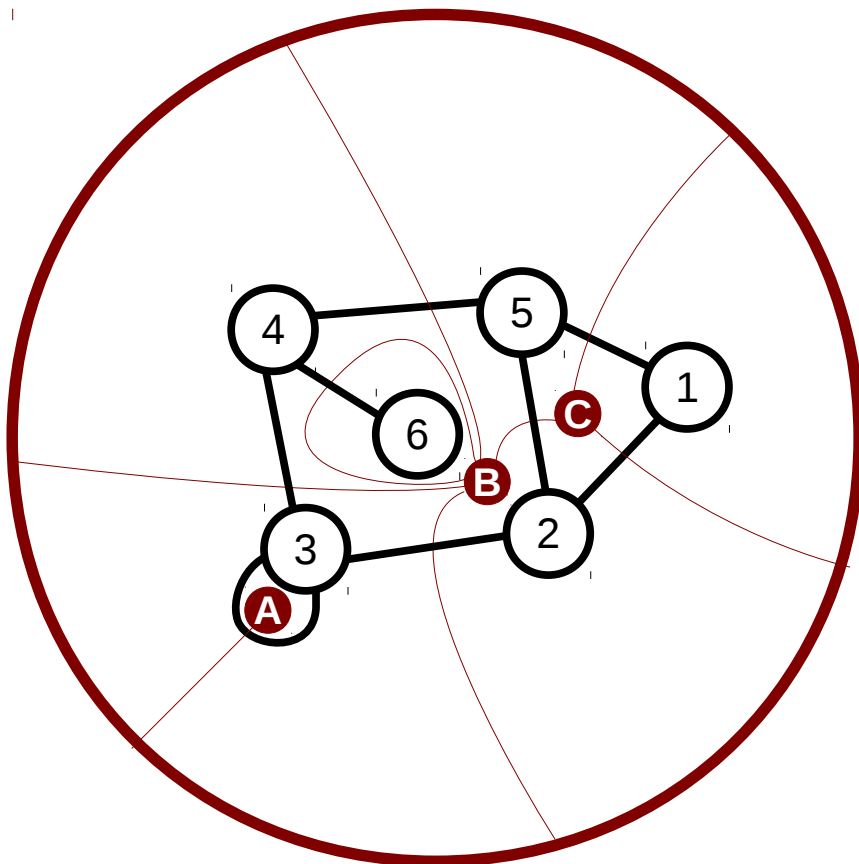
$$\{V : \{1, 2, 3, 4, 5, 6\},$$
$$E : \{\{1,2\}, \{2,3\}, \{2,5\}, \{3,3\},$$
$$\{4,3\}, \{4,5\}, \{4,6\}, \}$$
$$\rightarrow \{V : \{A, B, C, \infty\},$$
$$E : \{\{A, \infty\}, \{B, \infty\}^3, \{B, C\}, \{C, \infty\}^2,$$
$$\{\infty, \infty\}\}$$

2. Zut, c'est même pas une opération interne → multiensemble

1. « Face englobante » → noeud ∞
→ ∞ situé aux antipodes d'une sphère
(dite « de Riemann »)

La dualité n'est pas un calcul !

→ deux « *mêmes* » graphes n'ont pas « *même* » dual !



$$\begin{aligned} &V : \{1, 2, 3, 4, 5, 6\}, \\ &E : \{\{1,2\}, \{2,3\}, \{2,5\}, \{3,3\}, \\ &\quad \{4,3\}, \{4,5\}, \{4,6\}, \} \\ &\rightarrow \{V : \{A,B,C,\infty\}, \\ &E : \{\{A,\infty\}, \{B,\infty\}^3, \{B,C\}, \{C,\infty\}^2, \\ &\quad \{\infty,\infty\}\} \\ &\rightarrow \{V : \{A,B,C,\infty\}, \\ &E : \{\{A,\infty\}, \{B,B\}, \{B,\infty\}^3, \{B,C\}, \\ &\quad \{C,\infty\}^2, \} \\ &\{\infty,\infty\} \ll \text{est devenu} \gg \{B,B\} \end{aligned}$$

Quand il est impossible de calculer sur une notion, elle ne sert à rien.

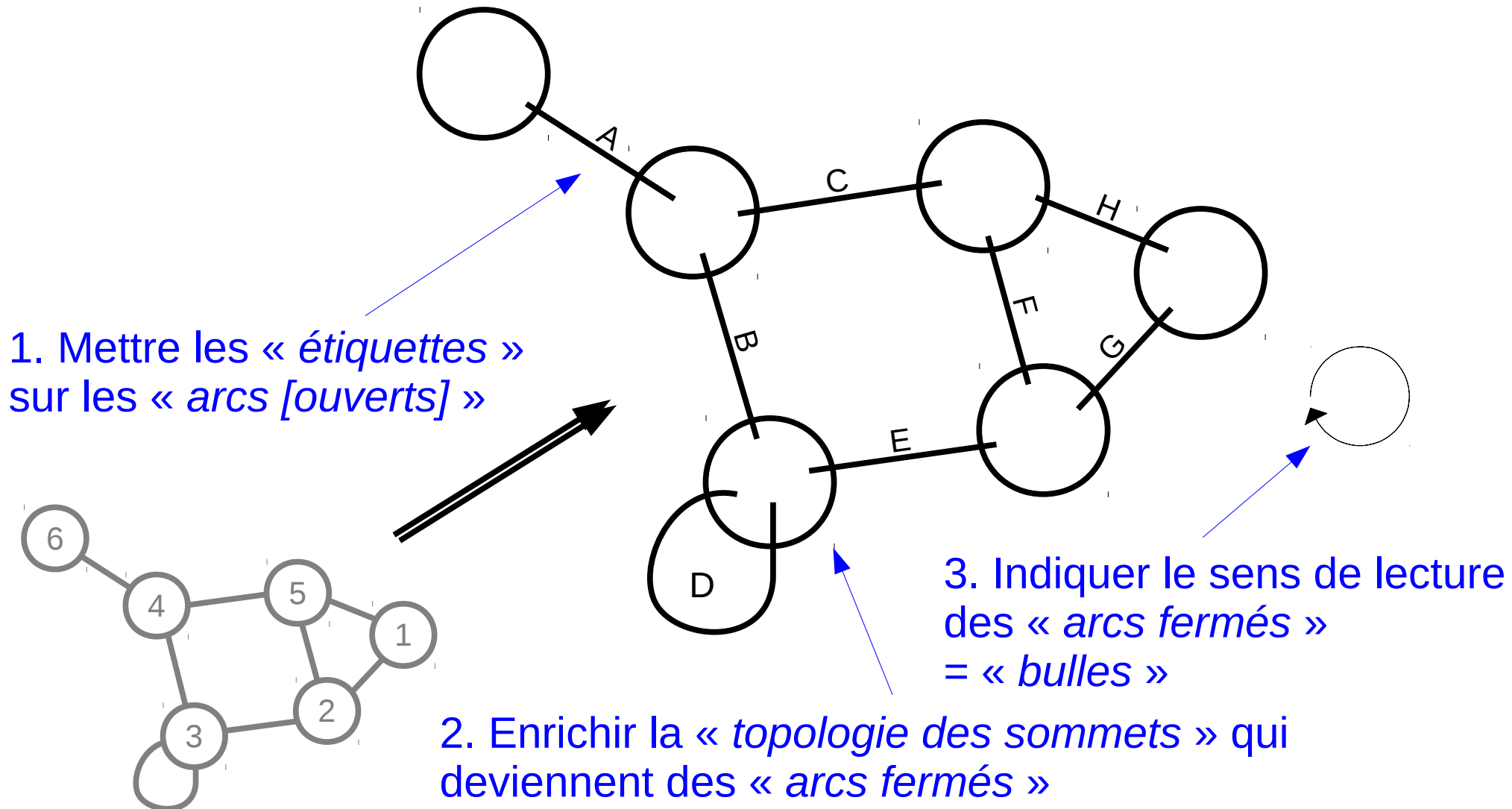
Créer une topologie calculable

- Permettant de traiter la dualité comme une opération calculable :
 - Lorsque les diagrammes ont des duals différents, ils ont des écritures différentes
 - Lorsque les diagrammes ont le même dual, ils ont la même écriture
- Comment « *augmenter* » la représentation des graphes de façon à obtenir cette propriété ?



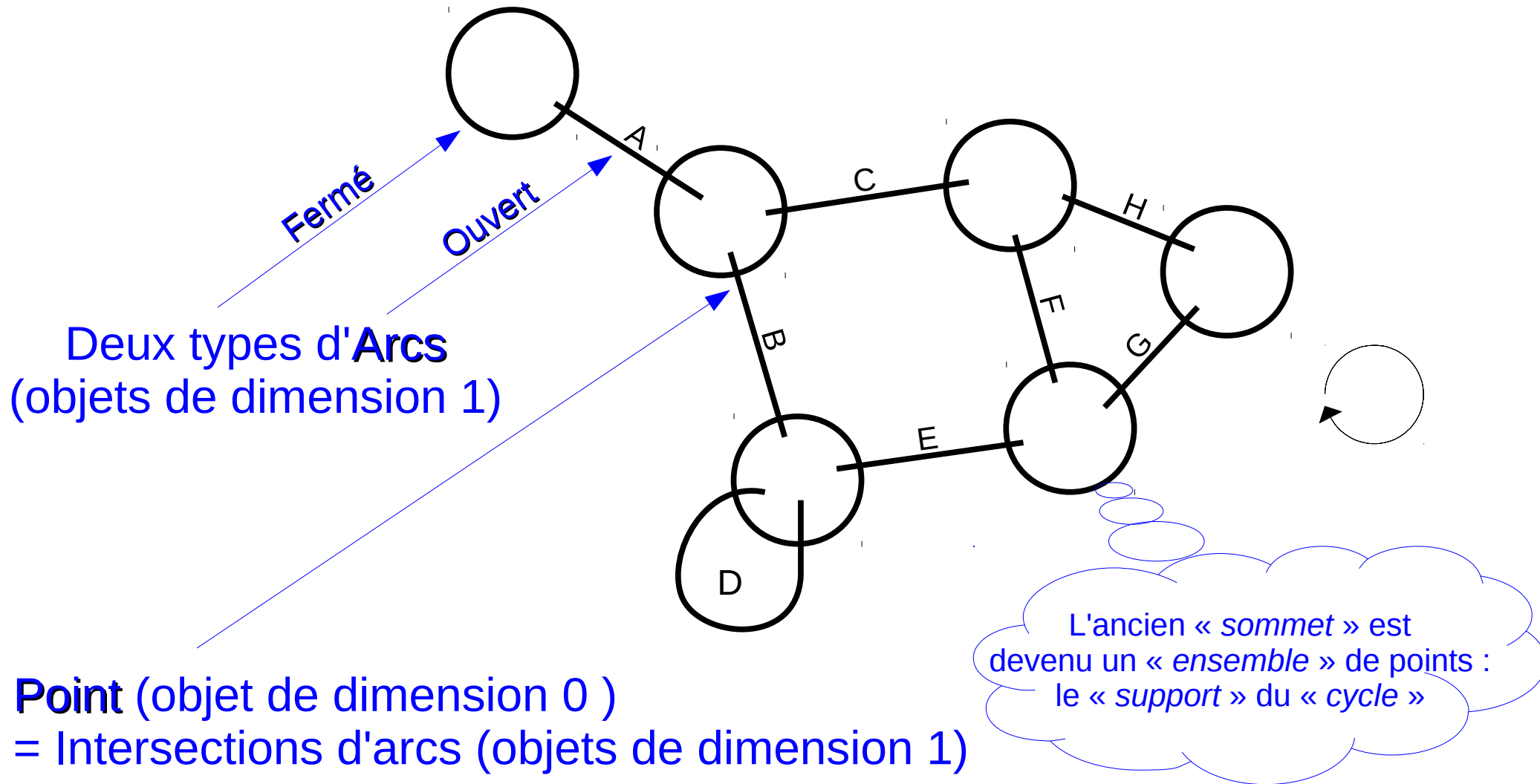
Une topologie comme
« *variété algébrique* »

La transformation « *modale* »



Déjà, de la robustesse !

d'un point de vue « topologique »



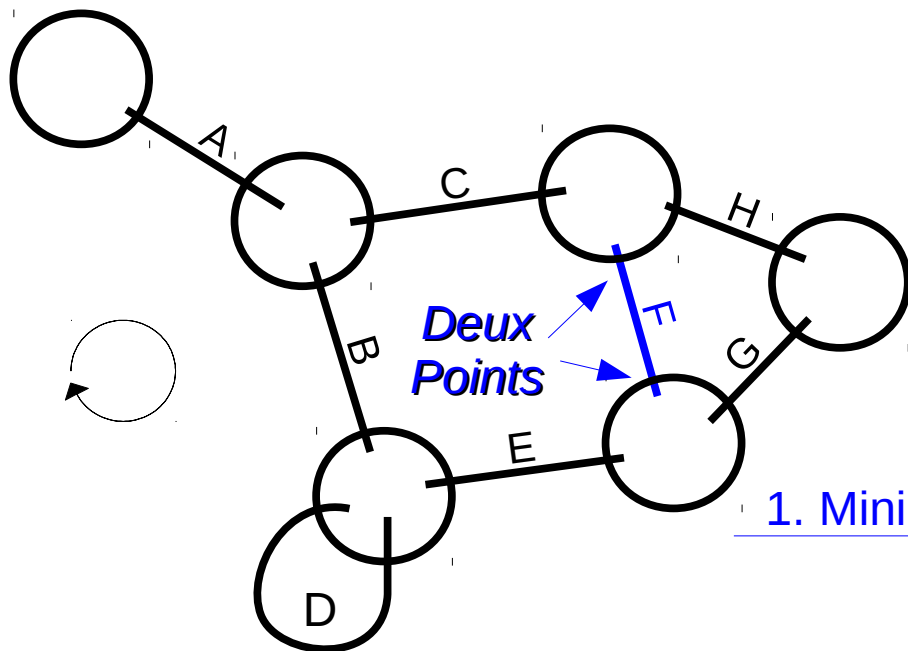
Une nouvelle écriture des graphes

{ } : Ensemble [de « *Cycles de symboles* »]

>_> : **Cycle**

[A-Z]* : **Symbole** = « signe présent en **deux occurrences** »

→ Ordonné selon l'ordre lexicographique



=

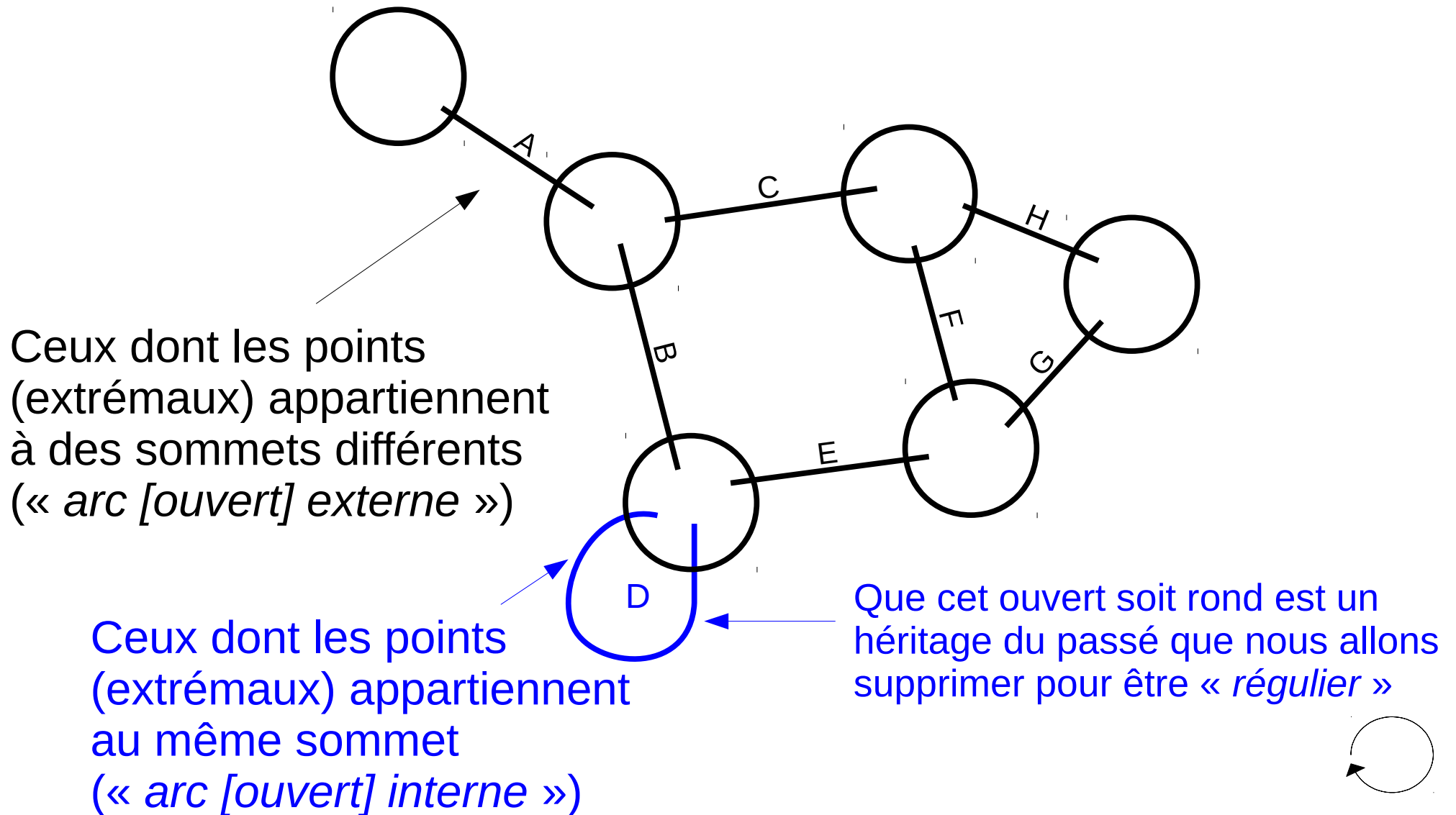
{
>A, >,
>A, C, B, >,
>B, E, D, D, >,
>C, H, **F**, >,
>E, **F**, G, >,
>G, H, >,
}

2. Tri lexicographique

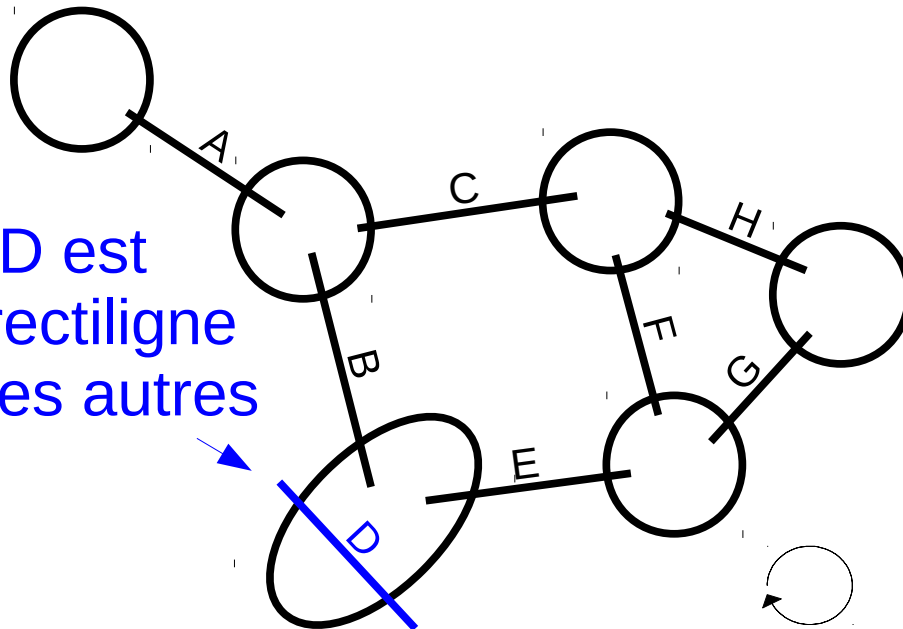
1. Minimum en premier

deuxième occurrence

Deux types d'« arcs ouverts »



Arcs [ouverts] « réguliers »



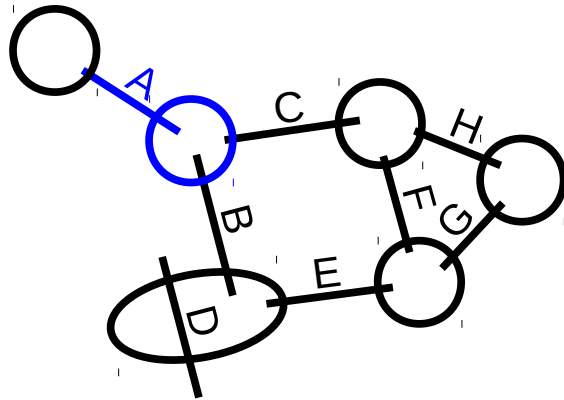
```

{
>A, >,
>A, C, B, >,
>B, E, D, D, >,
>C, H, F, >,
>E, F, G, >,
>G, H, >,
}

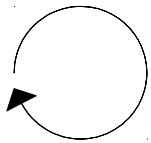
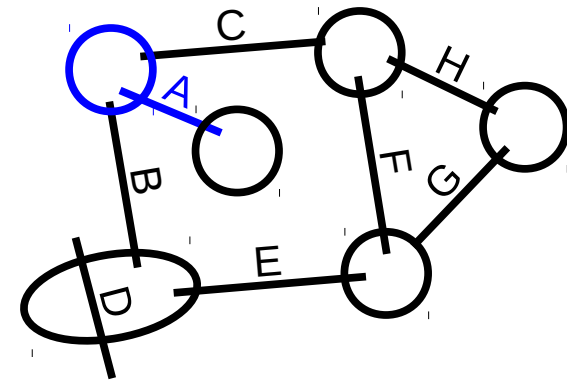
```

Attention : D ne doit ni couper entre B et E, ni être de « l'autre coté »
 → sinon cela change l'écriture (>B, D, E, D,> >B, D, D, E,>)

Compatibilité avec le dessin...



\neq

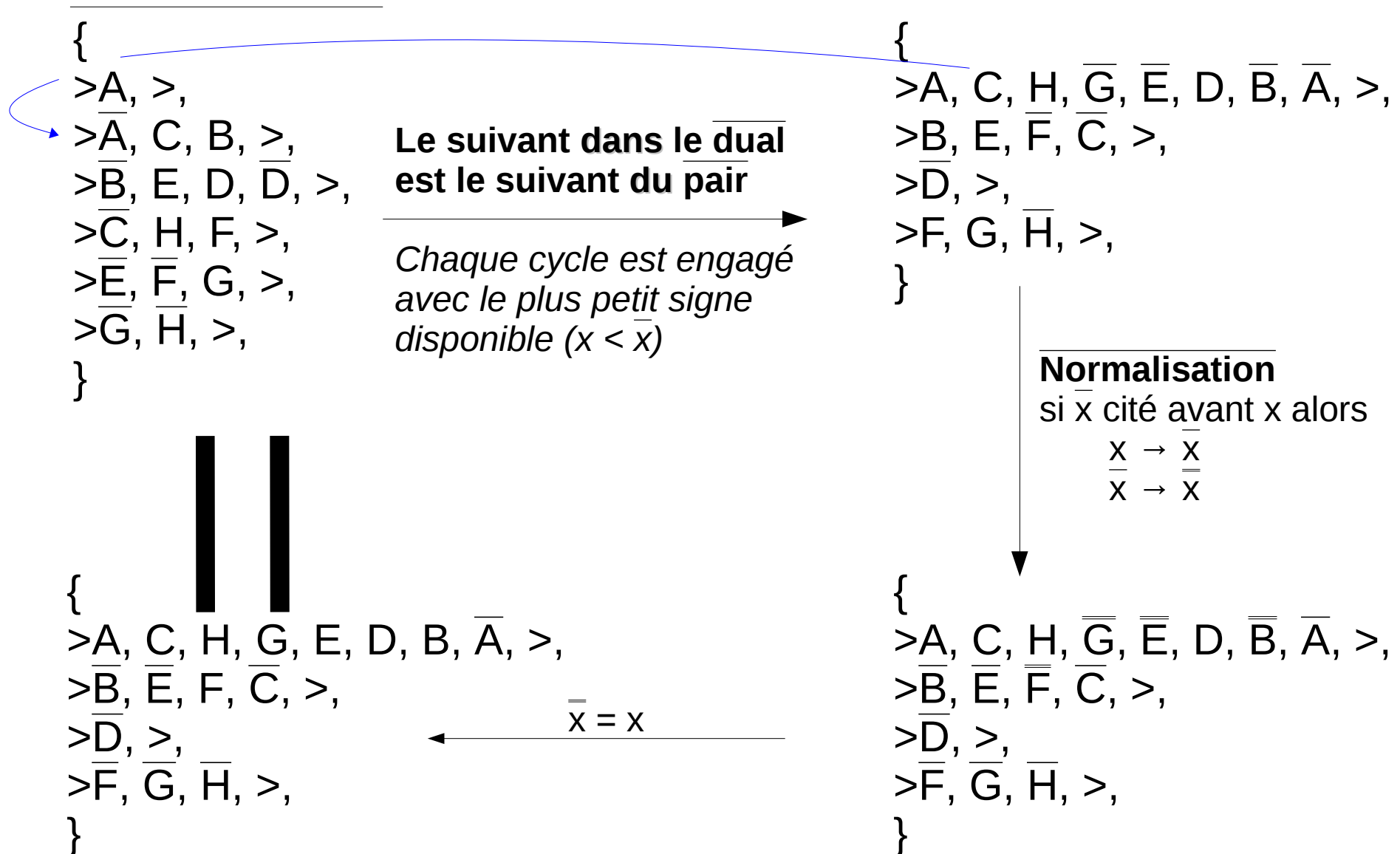


$\{$
 $>A, >,$
 $>A, C, B, >,$
 $>B, E, D, D, >,$
 $>C, H, F, >,$
 $>E, F, G, >,$
 $>G, H, >,$
 $\}$

\neq

$\{$
 $>A, >,$
 $>A, B, C, >,$
 $>B, E, D, D, >,$
 $>C, H, F, >,$
 $>E, F, G, >,$
 $>G, H, >,$
 $\}$

La dualité « *modale* » est un calcul



Convolution : Le dual du dual est le diagramme d'origine.

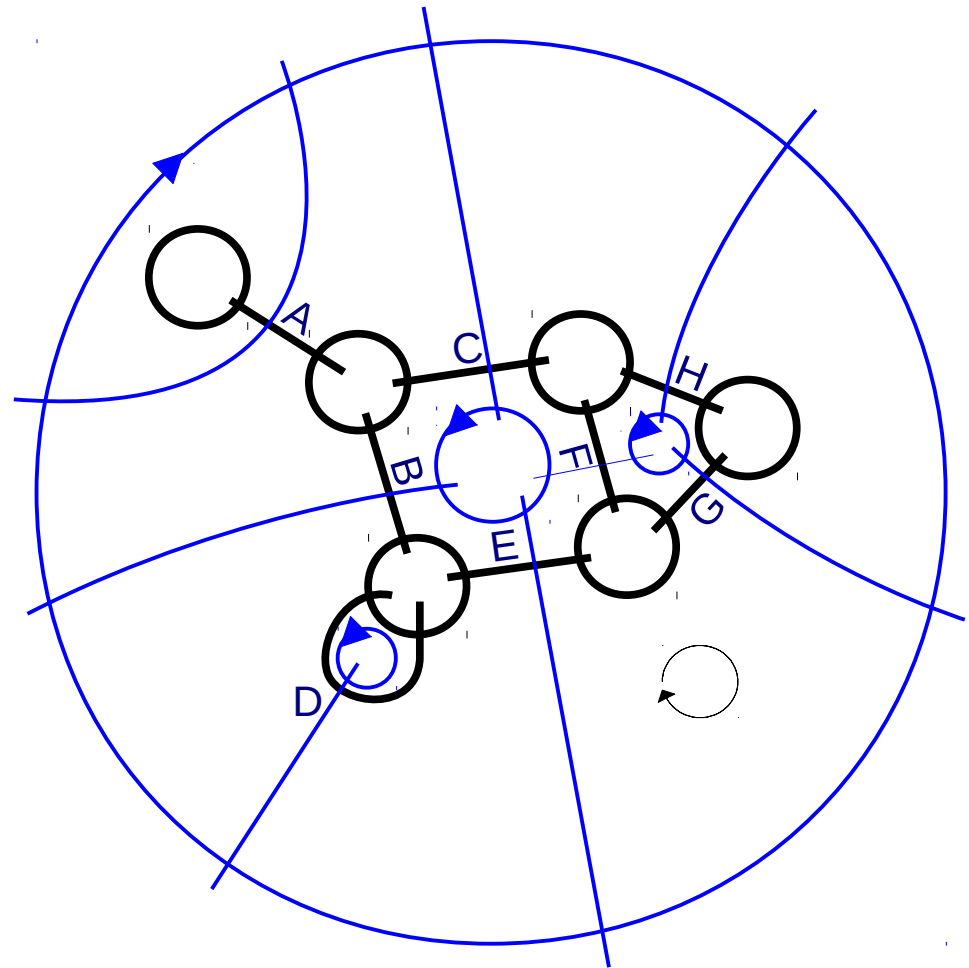
La dualité modale est une extension

```

{
>A, >,
>A, C, B, >,
>B, E, D, D, >,
>C, H, F, >,
>E, F, G, >,
>G, H, >,
}
    
```

```

||
{
>A, C, H, G, E, D, B, A, >,
>B, E, F, C, >,
>D, >,
>F, G, H, >,
}
    
```



de la dualité « *classique* ».

Une topologie calculable

- **Plus proche de le topologie du plan :**
 - Arcs ouverts, arcs fermés
 - Points comme intersections d'arcs
 - Écriture sur les arcs ouverts
 - « *ligne de base* » typographique
- **Capable de réaliser des calculs :**
 - Tout diagramme peut être réduit à une expression textuelle unique
 - *Égalité sémantique des diagrammes par égalité textuelle de la forme réduite (= « variété algébrique »)*
 - La dualité devient un calcul
 - *Complexité séquentielle $n.\log(n)$*

Plan de l'exposé

- Créer une topologie calculable
- **Traiter les données semi-structurées**
 - Transformation modale & Dualité
 - Restitution & Parcours
- Intégration du temps
- Conclusion & Perspectives

Un exemple d'XML...

<http://www.jacksay.com/cours/dist/slides/xml-01.html#/>

```
<?xml version="1.0" encoding="utf-8"?>
```

Noeud document

```
<personnages><
```

« *Enregistrement* » avec un attribut d'identification

```
<personnage id="42">
```

```
<pseudo>Le gouverneur</pseudo>
```

```
<prenom>Phillip</prenom>
```

```
<nom>Blake</nom>
```

```
<bio>
```

```
<p>Il gère Woodbury
```

```
</p>
```

```
<p>Il est vraiment
```

```
<strong>très</strong>
```

```
méchant
```

```
</p>
```

```
</bio>
```

```
</personnage>
```

```
<!-- autres personnages -->
```

```
<personnage>...</personnages>
```

```
<personnage>...</personnages>
```

```
</personnages>
```

Noeuds avec un nœud texte simple

Noeud avec un nœud texte riche

... orienté base de données

Variable d'extension

Non formalisée : erreurs de syntaxe

Doté d'une représentation graphique

<http://www.jacksay.com/cours/dist/slides/xml-01.html#/>

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<personnages>
```

```
  <personnage id="42">
```

```
    <pseudo>Le gouverneur</pseudo>
```

```
    <prenom>Phillip</prenom>
```

```
    <nom>Blake</nom>
```

```
    <bio>
```

```
      <p>Il gère Woodbury
```

```
    </p>
```

```
    <p>Il est vraiment
```

```
      <strong>très</strong>
```

```
      méchant
```

```
    </p>
```

```
  </bio>
```

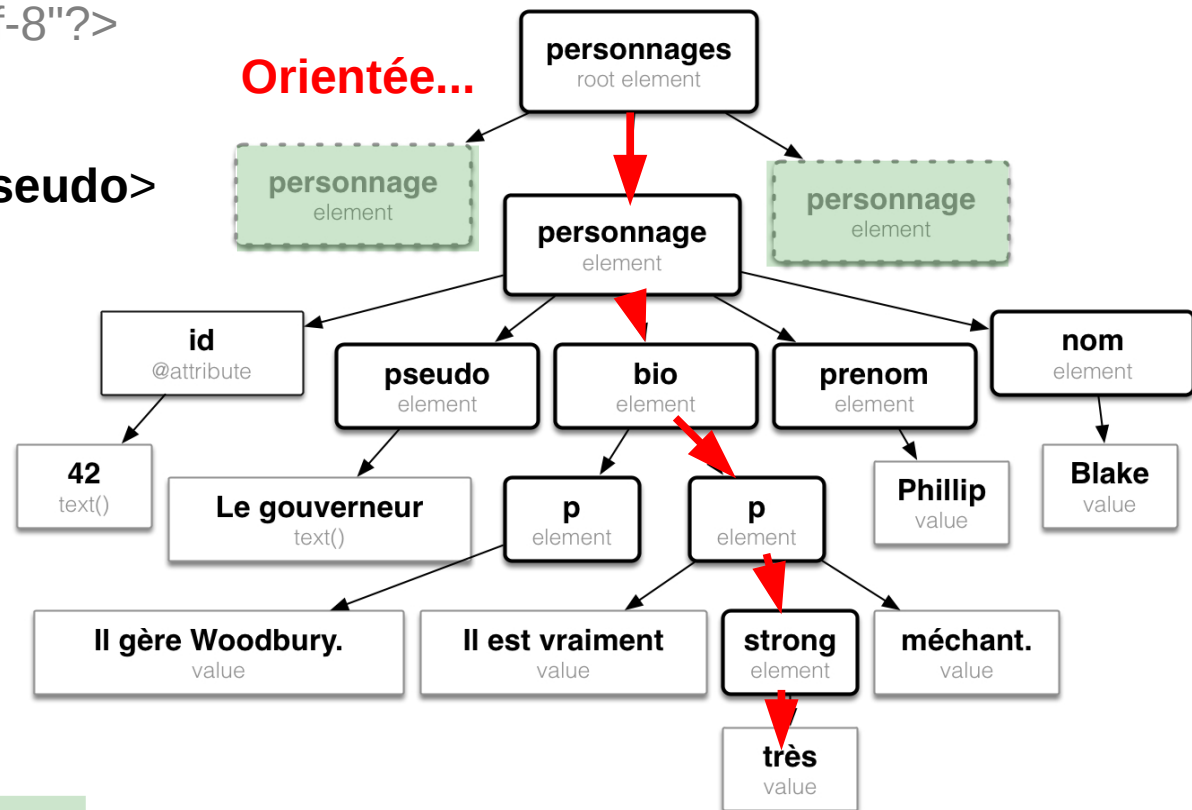
```
</personnage>
```

```
<!-- autres personnages -->
```

```
<personnage>...</personnage>
```

```
<personnage>...</personnage>
```

```
</personnages>
```



Mais pas ordonnée...

La transformation « *modale* »

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<personnages>
```

```
  <personnage id="42">
```

```
    <pseudo>Le gouverneur</pseudo>
```

```
    <prenom>Phillip</prenom>
```

```
    <nom>Blake</nom>
```

```
    <bio>
```

```
      <p>Il gère Woodbury
```

```
    </p>
```

```
    <p>Il est vraiment
```

```
      <strong>très</strong>
```

```
      méchant
```

```
    </p>
```

```
  </bio>
```

```
</personnage>
```

```
</personnages>
```

1. Identifier les nœuds
(DOM)

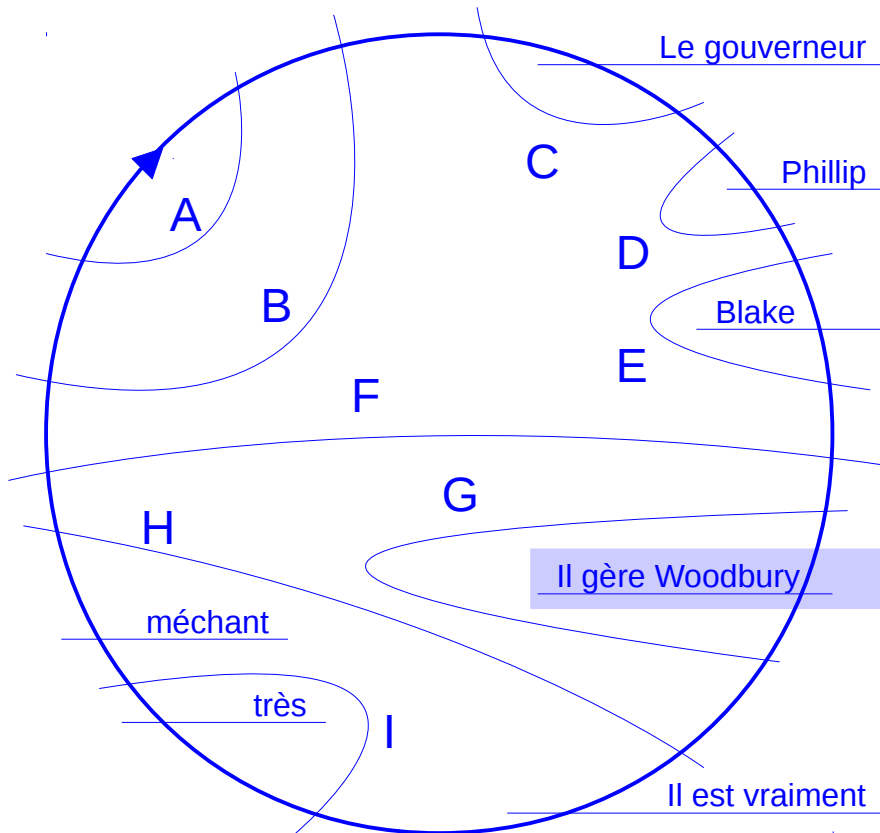
3. Nœud texte «stripped»

2. Recopier avec le tag fermant
comme pair du tag ouvrant
(SAX)

A : personnages
B : personnage id="42"
C : pseudo
D : prenom
E : nom
F : bio
G : p
H : p
I : strong

```
{  
>A, B, C, "Le gouverneur",  $\bar{C}$ , D,  
"Phillip",  $\bar{D}$ , E, "Blake",  $\bar{E}$ , F, G,  
"Il gère Woodbury",  $\bar{G}$ , H, "Il est  
vraiment", I, "très",  $\bar{I}$ , "méchant",  
 $\bar{H}$ ,  $\bar{F}$ ,  $\bar{B}$ ,  $\bar{A}$ , >,  
}
```

Équivalence hyperbolique



```
<?xml version="1.0" encoding="utf-8"?>
<personnages>
  <personnage id="42">
    <pseudo>Le gouverneur</pseudo>
    <prenom>Phillip</prenom>
    <nom>Blake</nom>
    <bio>
      <p>Il gère Woodbury</p>
      <p>Il est vraiment
        <strong>très</strong> méchant</p>
    </bio>
  </personnage>
</personnages>
```

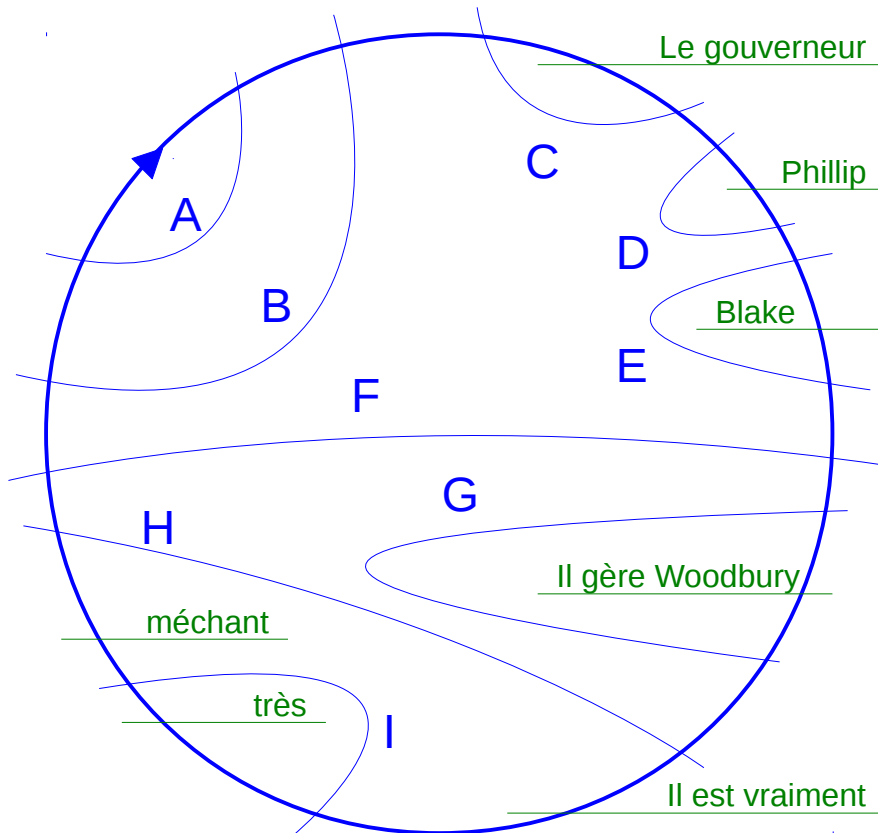
```
{
  >A, B, C, "Le gouverneur", C̄, D, "Phillip",
  D̄, E, "Blake", Ē, F, G, "Il gère Woodbury",
  Ḡ, H, "Il est vraiment", I, "très", Ī,
  "méchant", H̄, F̄, B̄, Ā, >,
}
```

```
A : personnages
B : personnage id="42"
C : pseudo
D : prenom
E : nom
F : bio
G : p
H : p
I : strong
```

Topologie semi-structurée

- La formulation modale est une extension de la théorie des graphes standard (Cf. Partie 1)
- La formulation modale est une extension des données semi-structurée (« *Equivalence hyperbolique* »)
- Toujours dans l'objectif d'assurer la robustesse de l'approche face au calcul, nous allons calculer le dual du graphe hyperbolique.

Calculons le dual



Le suivant dans le $\overline{\text{dual}}$ est le suivant du pair

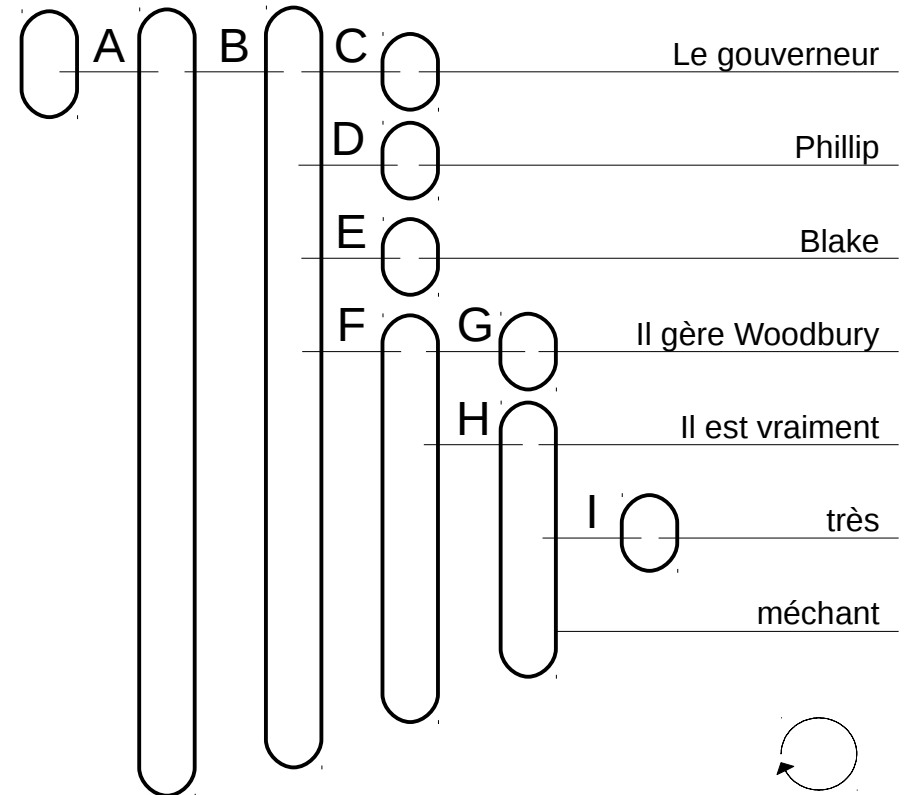
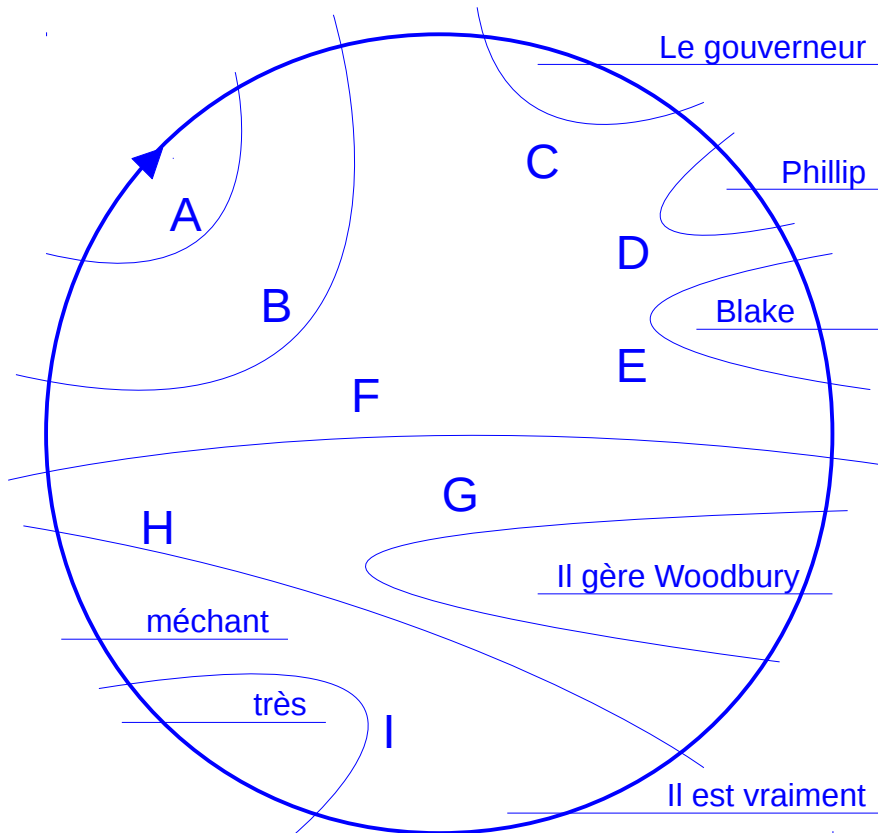
Chaque cycle est engagé avec le plus petit signe disponible ($x < \bar{x}$)

Tout nœud texte est son Propre dual : " $_$ " = " $\bar{_}$ "

```
{
>A, B, C, "Le gouverneur",  $\bar{C}$ , D, "Phillip",
 $\bar{D}$ , E, "Blake",  $\bar{E}$ , F, G, "Il gère Woodbury",
 $\bar{G}$ , H, "Il est vraiment", I, "très",  $\bar{I}$ ,
"méchant",  $\bar{H}$ ,  $\bar{F}$ ,  $\bar{B}$ ,  $\bar{A}$ , >,
}
```

```
{
>A, >,
> $\bar{B}$ ,  $\bar{A}$ , >,
> $\bar{B}$ , C, D, E, F, >,
> $\bar{C}$ , "Le gouverneur", >,
> $\bar{D}$ , "Phillip", >,
> $\bar{E}$ , "Blake", >,
> $\bar{F}$ , G, H, >,
> $\bar{G}$ , "Il gère Woodbury", >,
> $\bar{H}$ , "Il est vraiment", I, "méchant", >,
> $\bar{I}$ , "très", >,
}
```

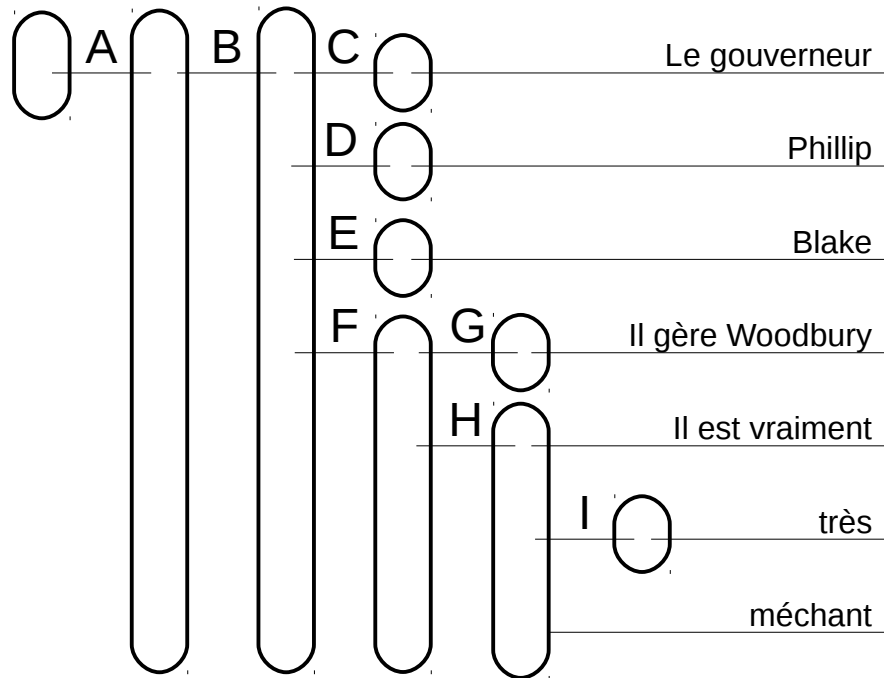
Représentons le dual



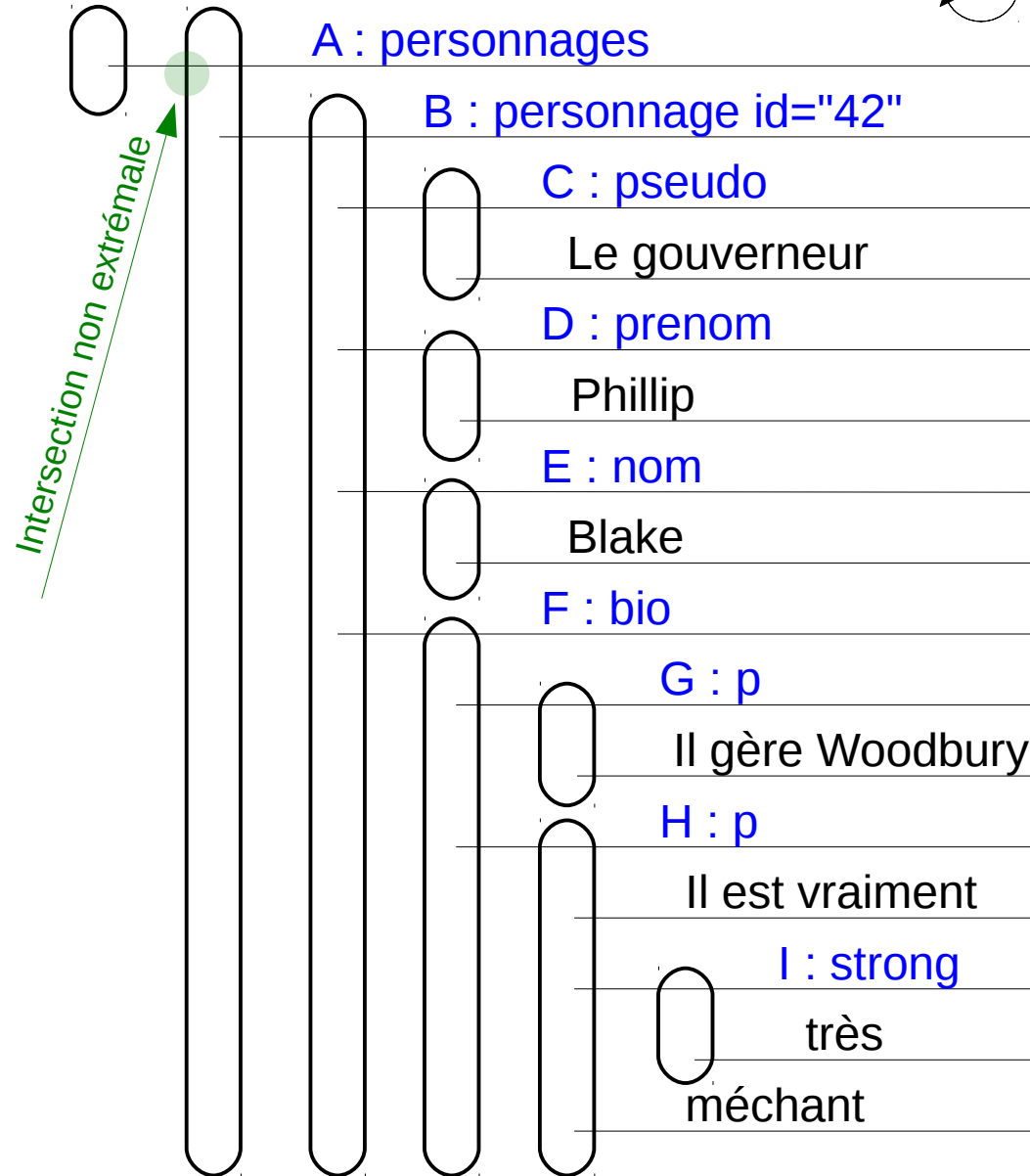
```
{
>A, B, C, "Le gouverneur",  $\bar{C}$ , D, "Phillip",
 $\bar{D}$ , E, "Blake",  $\bar{E}$ , F, G, "Il gère Woodbury",
G, H, "Il est vraiment", I, "très",  $\bar{I}$ ,
"méchant",  $\bar{H}$ ,  $\bar{F}$ ,  $\bar{B}$ ,  $\bar{A}$ , >,
}
```

```
{>A, >, >B,  $\bar{A}$ , >, > $\bar{B}$ , C, D, E, F, >,
> $\bar{C}$ , "Le gouverneur", >, > $\bar{D}$ , "Phillip", >,
> $\bar{E}$ , "Blake", >, > $\bar{F}$ , G, H, >,
>G, "Il gère Woodbury", >,
> $\bar{H}$ , "Il est vraiment", I, "méchant", >,
> $\bar{I}$ , "très", >, }
```

Réinsérons la table des noeuds



A : personnages
 B : personnage id="42"
 C : pseudo
 D : prenom
 E : nom
 F : bio
 G : p
 H : p
 I : strong



Le dual restitue la forme textuelle !

```
<?xml version="1.0" encoding="utf-8"?>
<personnages>
  <personnage id="42">
    <pseudo>Le gouverneur</pseudo>
    <prenom>Phillip</prenom>
    <nom>Blake</nom>
    <bio>
      <p>Il gère Woodbury</p>
      <p>Il est vraiment
        <strong>très</strong>
        méchant
      </p>
    </bio>
  </personnage>
</personnages>
```

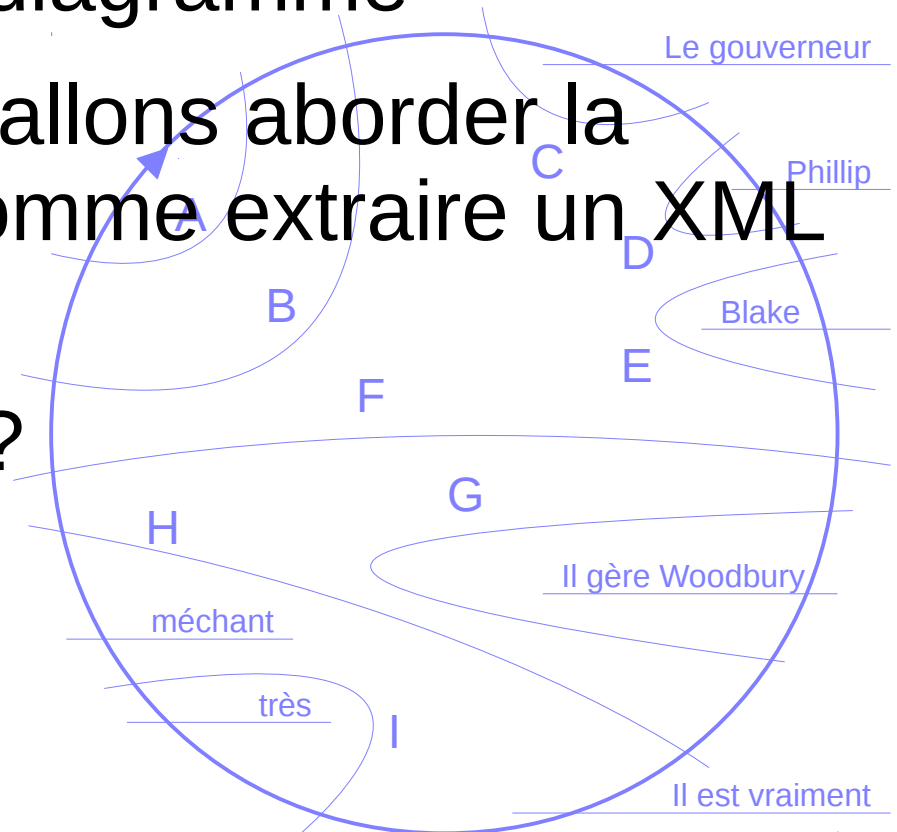
Diagram illustrating the mapping from XML to a flat text representation (the 'personnages' table). The XML structure is mapped to the table rows as follows:

- A**: `<pseudo>` maps to the start of the `pseudo` row.
- B**: `<prenom>` maps to the start of the `prenom` row.
- C**: `<nom>` maps to the start of the `nom` row.
- D**: `<bio>` maps to the start of the `bio` row.
- E**: `<p>` maps to the start of the first paragraph within the `bio` row.
- F**: `</p>` maps to the end of the first paragraph within the `bio` row.
- G**: `<p>` maps to the start of the second paragraph within the `bio` row.
- H**: `` maps to the start of the `strong` tag within the second paragraph.
- I**: `` maps to the end of the `strong` tag within the second paragraph.

| personnages | |
|--------------------|------------------|
| personnage id="42" | |
| | pseudo |
| | Le gouverneur |
| | prenom |
| | Phillip |
| | nom |
| | Blake |
| | bio |
| | p |
| | Il gère Woodbury |
| | p |
| | Il est vraiment |
| | strong |
| | très |
| | méchant |

Réciproque

- Dans la première phase nous avons montré comment n'importe quel fichier XML pouvait se représenter comme un diagramme
- Dans cette phase nous allons aborder la question réciproque : comment extraire un XML de tout diagramme ?
- Et à quelles conditions ?



Similitudes et potentiels

- Diagramme modal

- Tout nœud a un nœud **suivant**
- Tout nœud a un nœud pair
- Tout nœud est le **suivant** d'un nœud
- Tout nœud et le pair de son pair

- DOM (= loaded XLM)

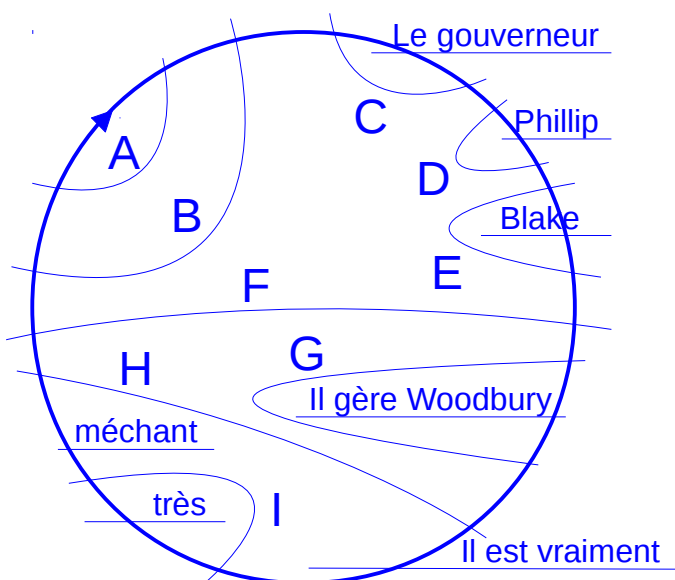
- Tout nœud **peut** avoir un **suivant**
- Tout nœud **peut** avoir un premier
- Tout nœud, sauf la racine, est soit le **suivant**, soit le premier d'un autre nœud
- La racine n'est le **suivant** ni le premier d'aucun nœud et n'a qu'un premier.

Cette **possibilité** d'avoir un **suivant** ou un **premier** va impliquer une **décision**.

Exploration de la similitude

- Diagramme modal

```
{
>A, B, C, "Le gouverneur", C̄, D,
"Phillip", D̄, E, "Blake", Ē, F, G,
"Il gère Woodbury", Ḡ, H, "Il est
vraiment", I, "très", Ī, "méchant",
H̄, F̄, B̄, Ā, >, }
```

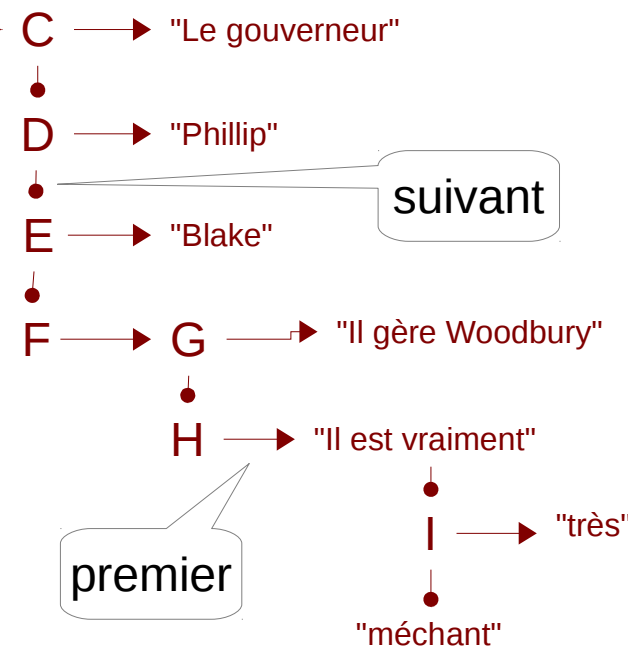


racine

A : personnages
 B : personnage id="42"
 C : pseudo
 D : prenom
 E : nom
 F : bio
 G : p
 H : p
 I : strong

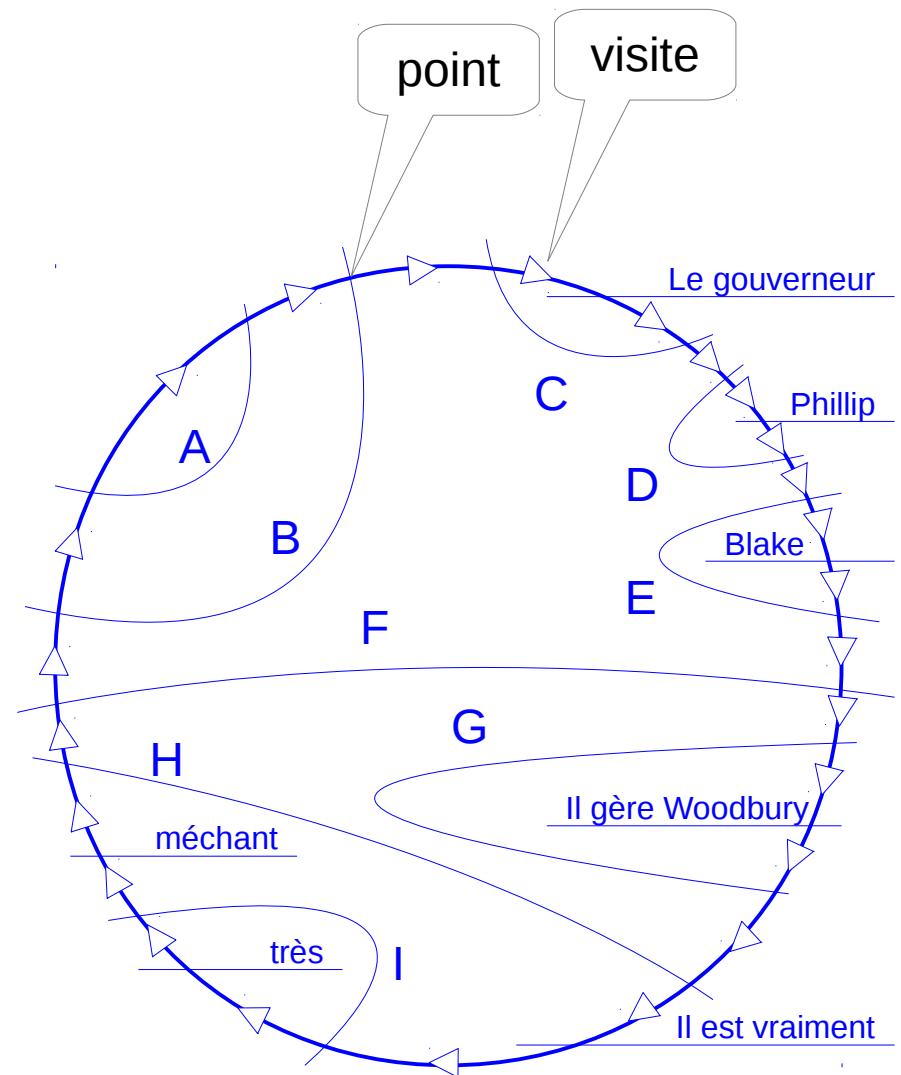
- Document Obj. Model

```
A(B(C("Le gouverneur"),
D("Phillip"),
E("Blake"),
F(
G("Il gère Woodbury"),
H("Il est vraiment", I("très"), "méchant")))))
```



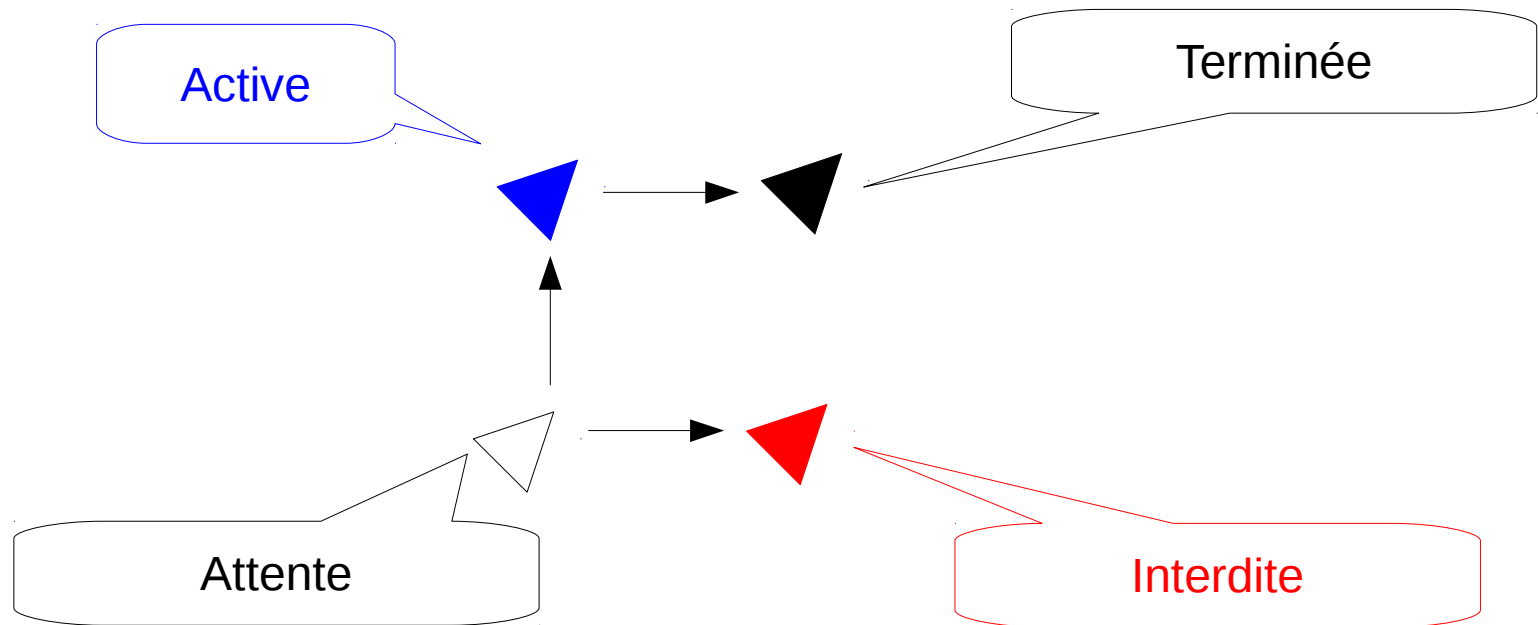
Visites

- Nous associons à chaque intervalle entre deux points un objet appelé **visite**
- Pour retrouver une structure XML, nous allons colorier les visites.

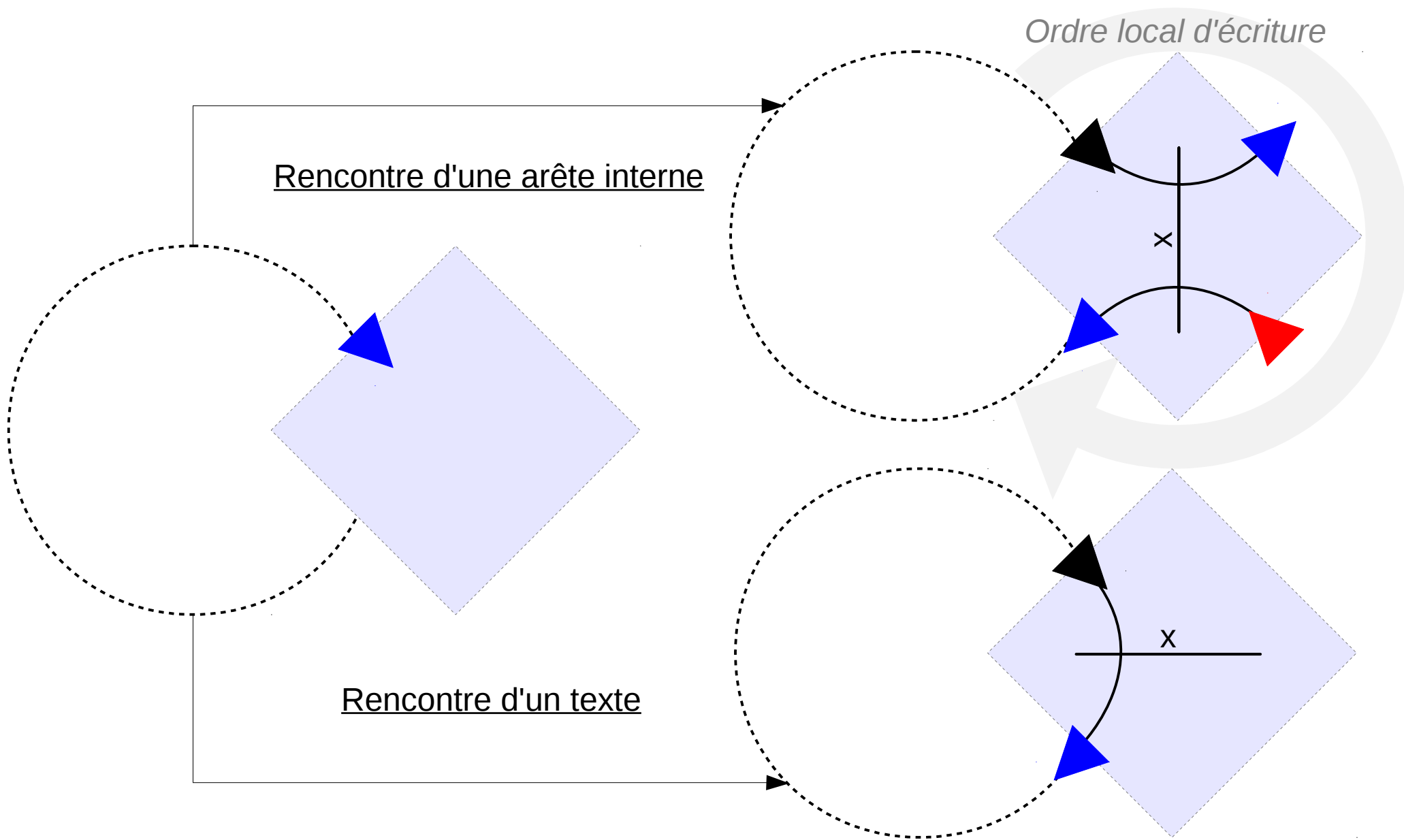


Comportement des visites

- Chaque visite à le comportement suivant :

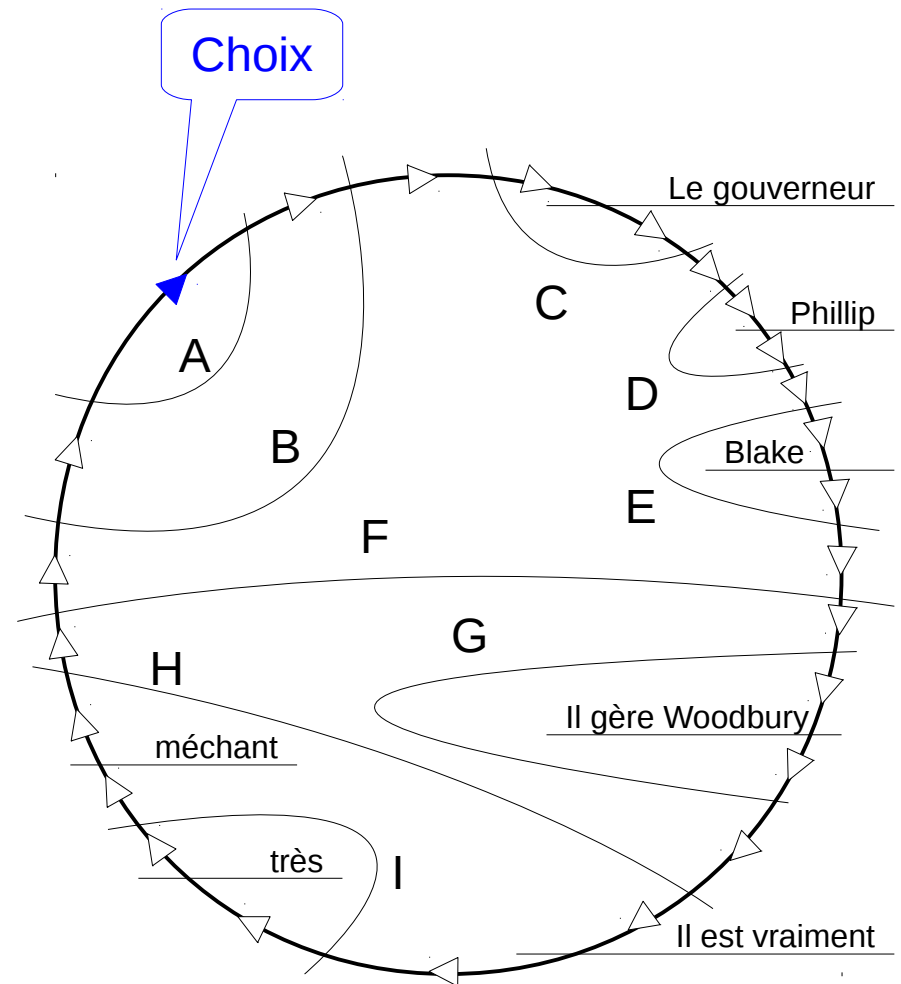


Coloriage des visites

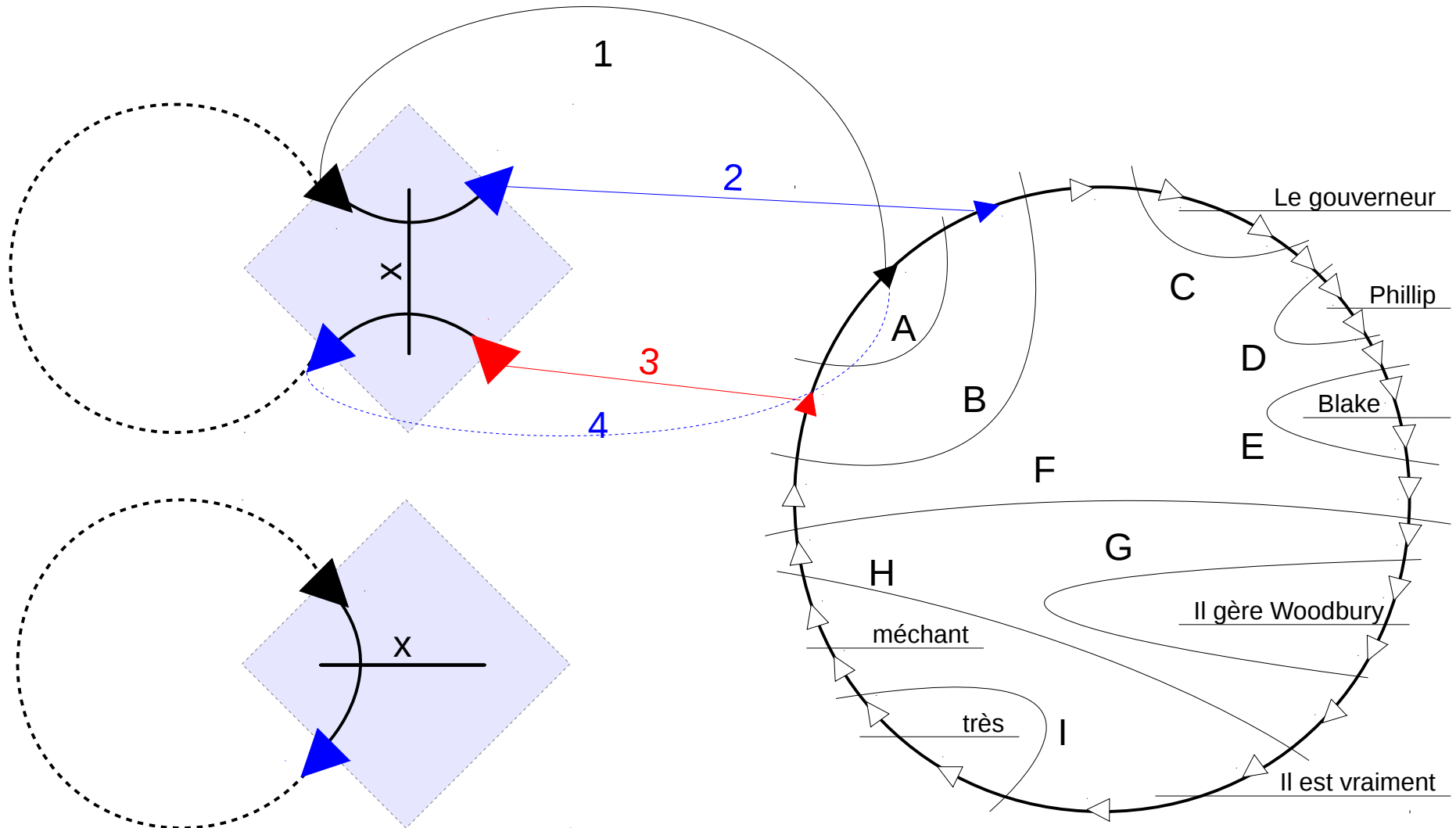


Initiation du coloriage des visites

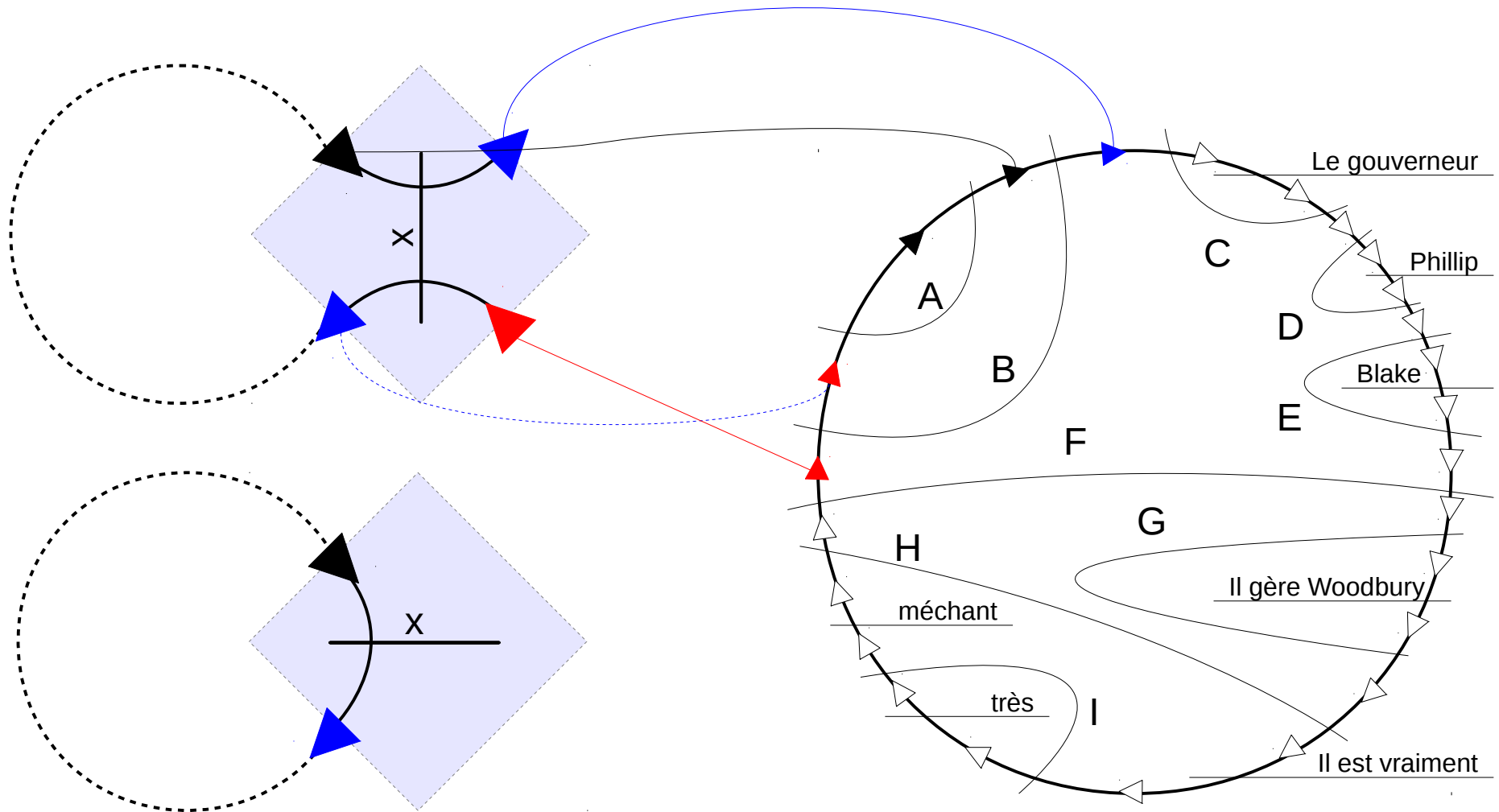
- Pour initier le coloriage, nous devons choisir une visite active.
- Ce choix va permettre de choisir premier et suivant pour toutes les visites



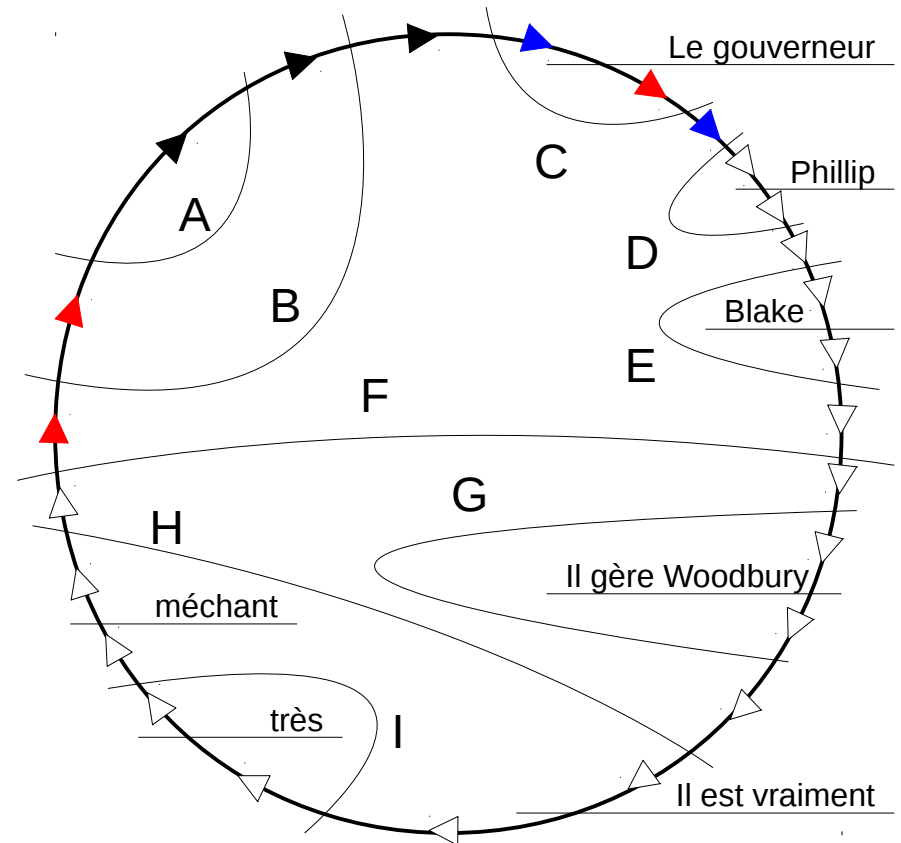
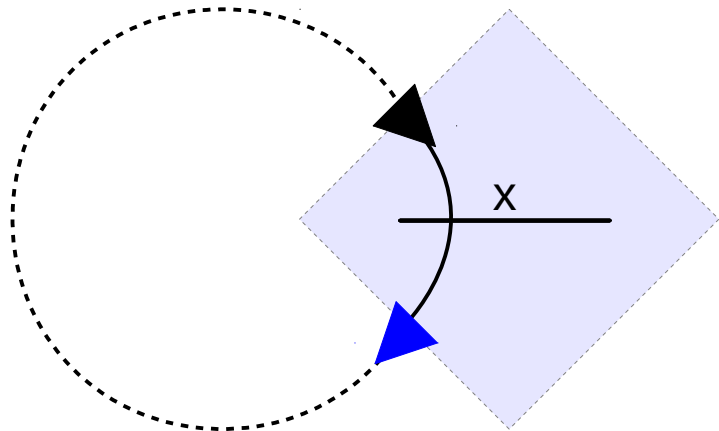
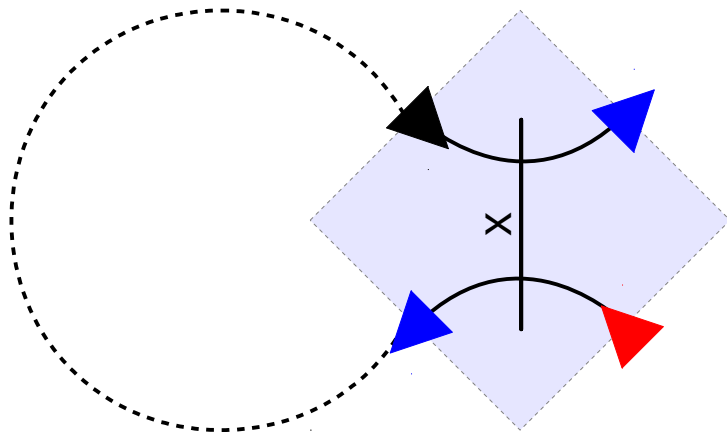
Coloriage des visites (1)



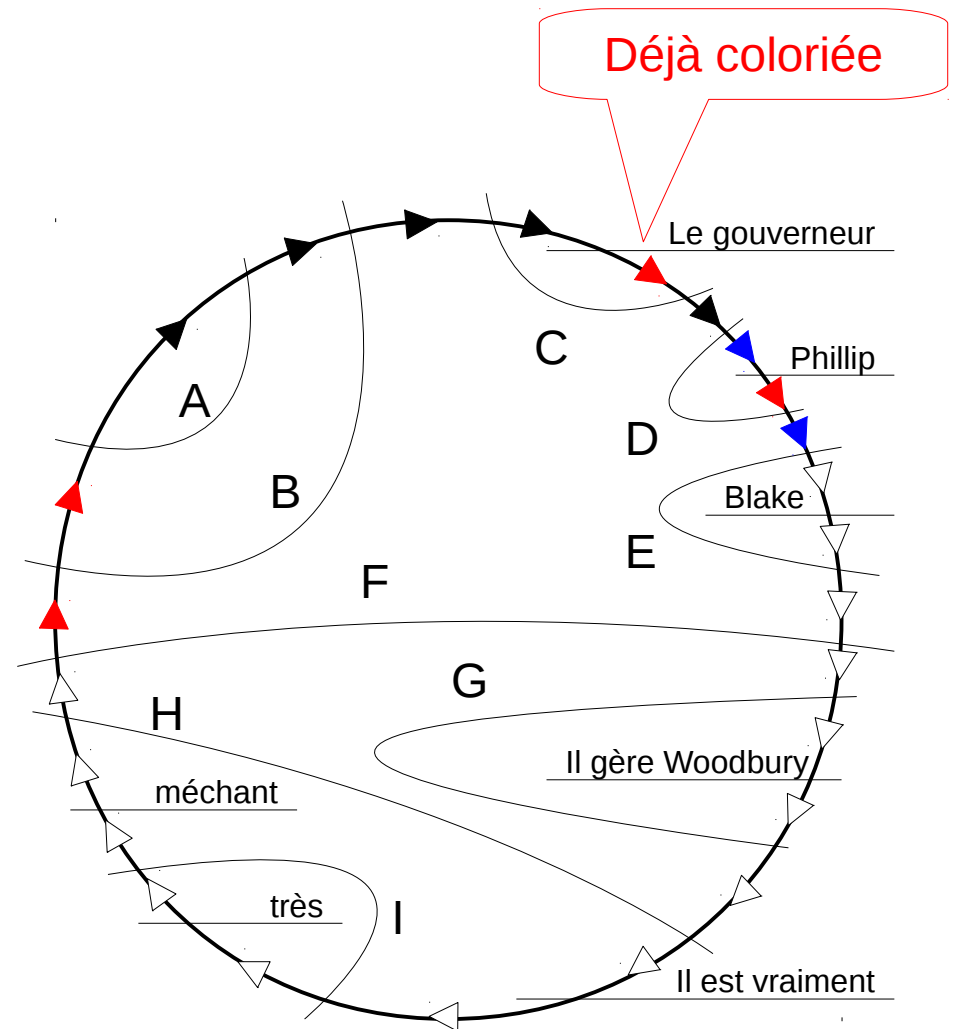
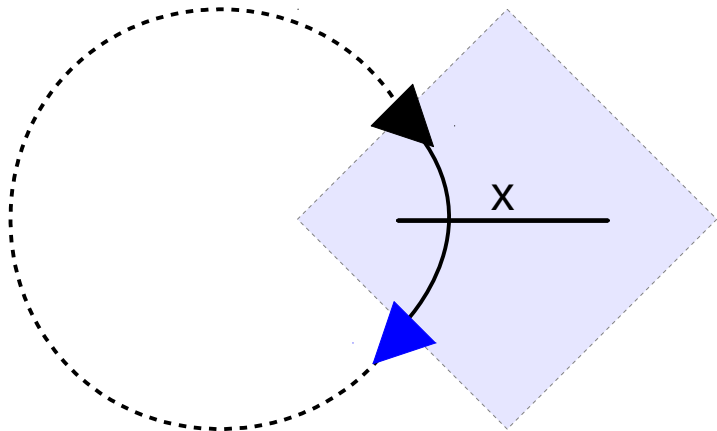
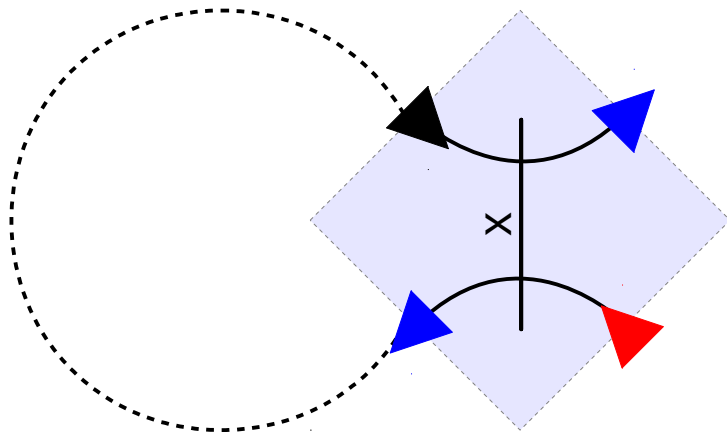
Coloriage des visites (2)



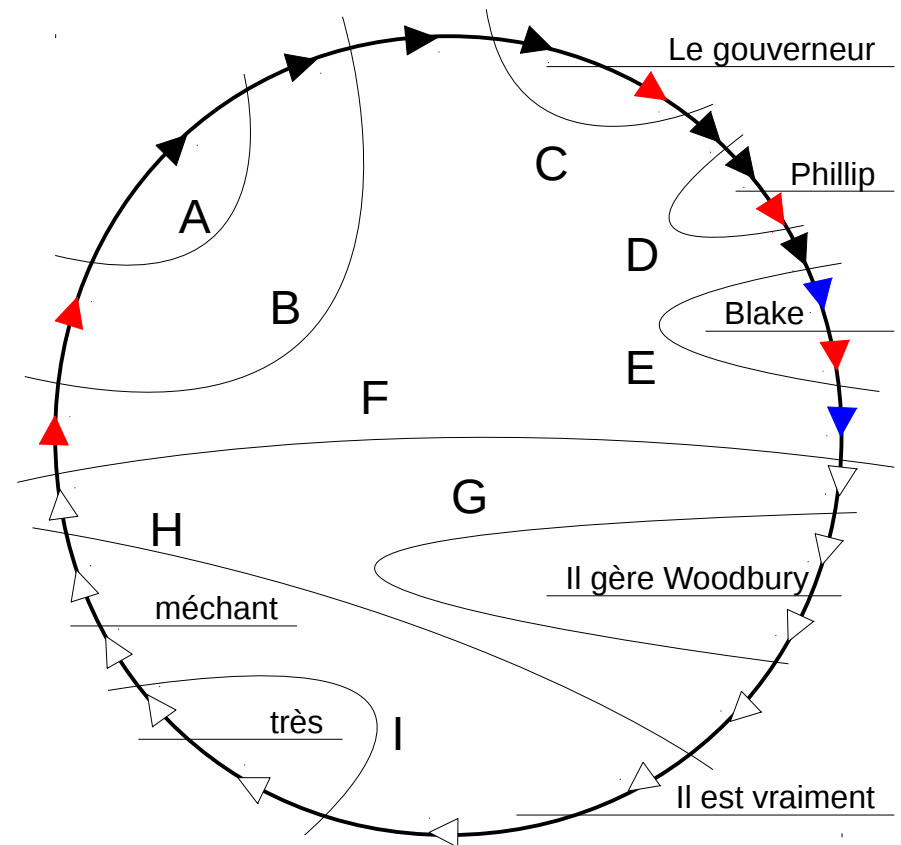
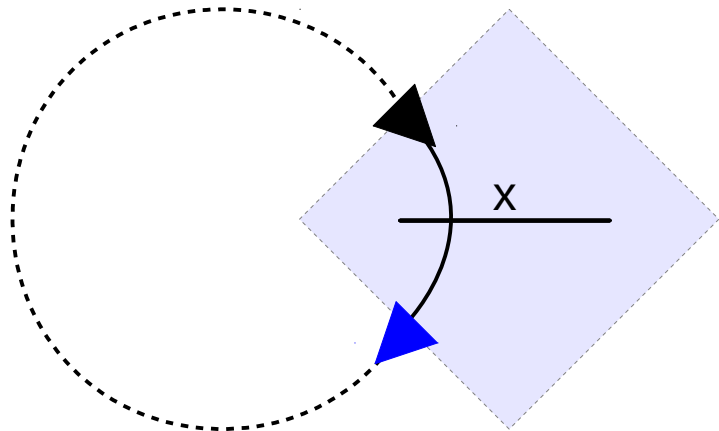
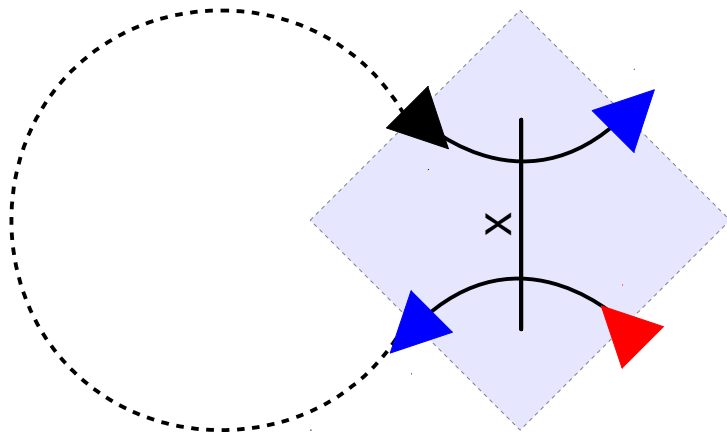
Coloriage des visites (3)



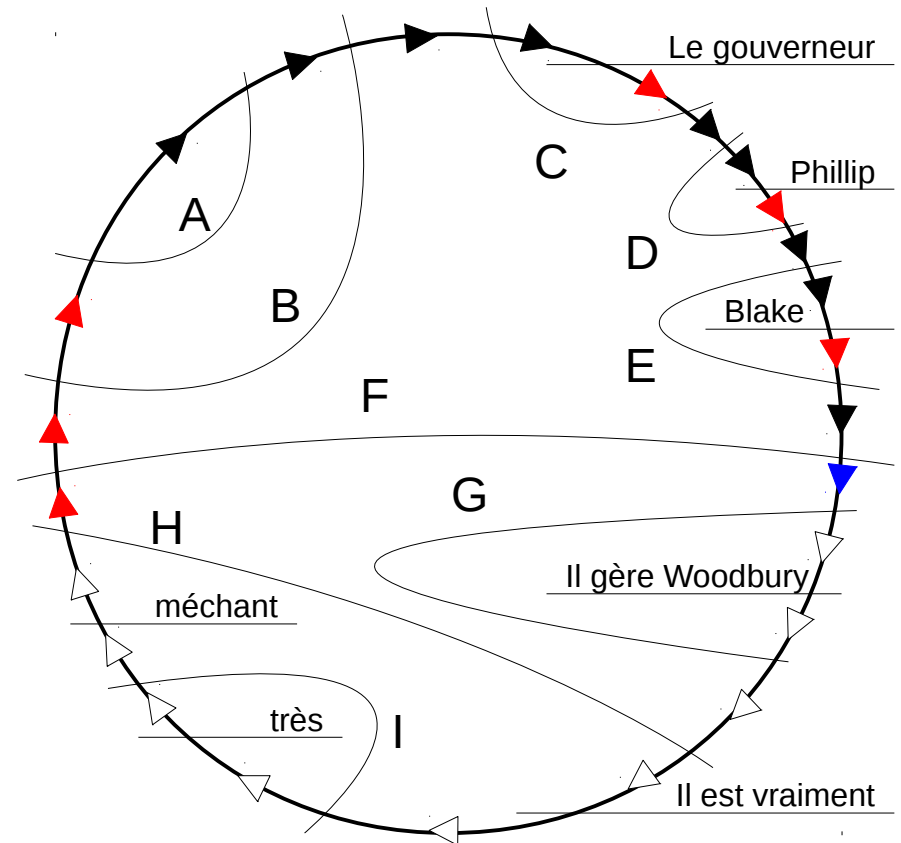
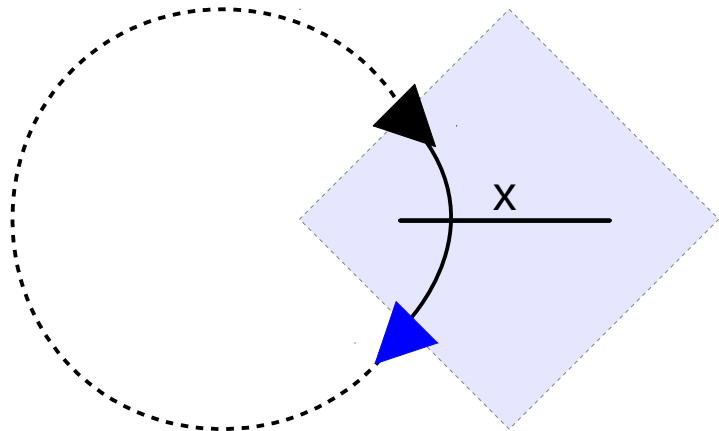
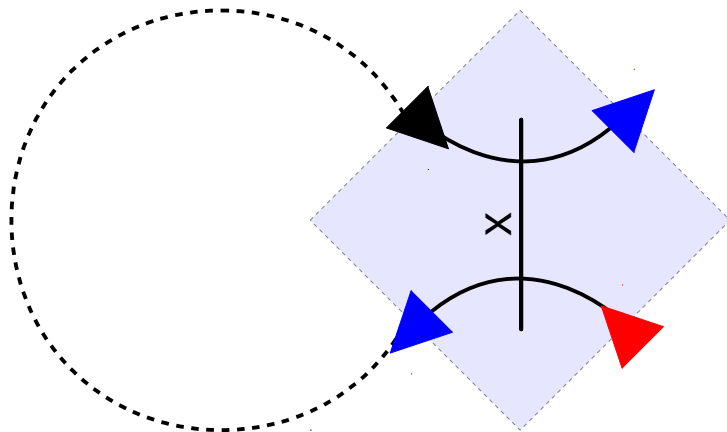
Coloriage des visites (4)



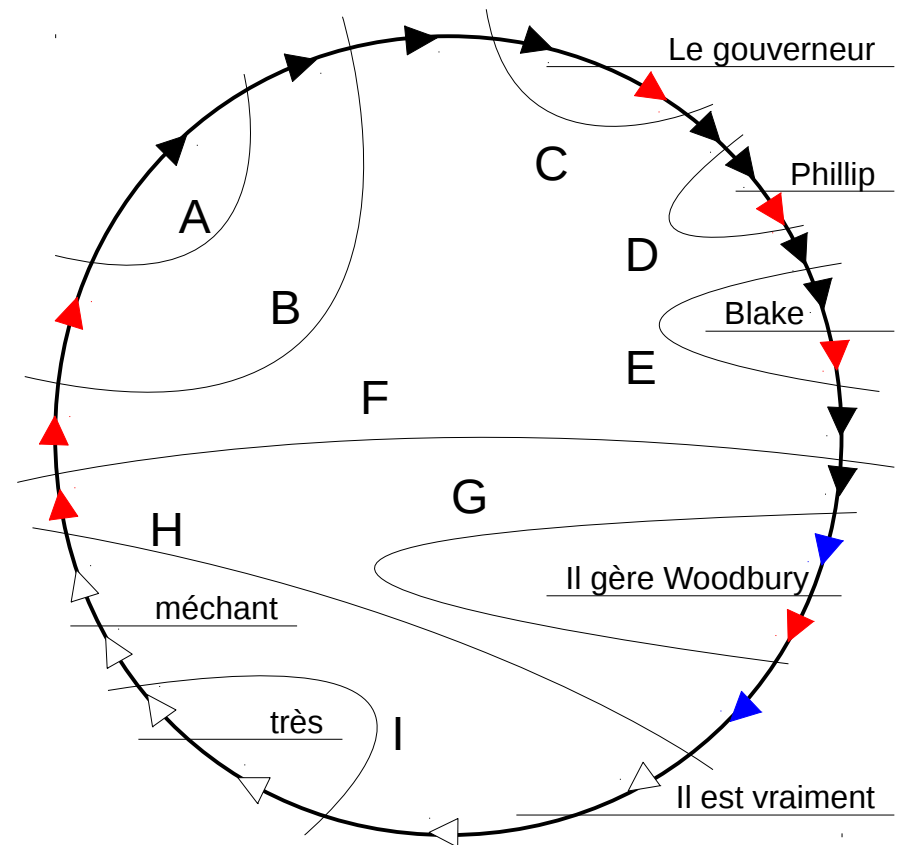
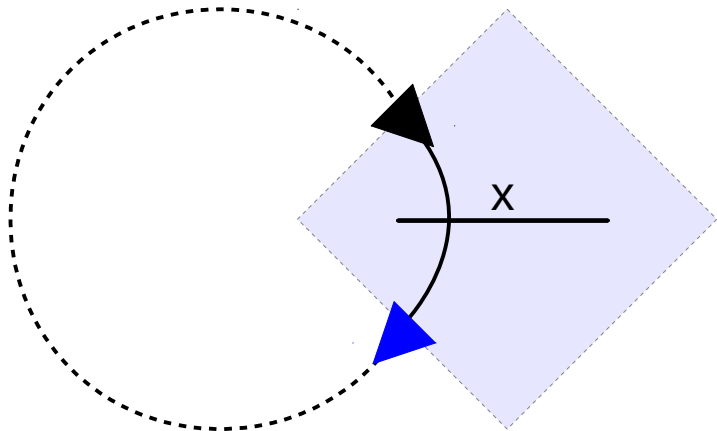
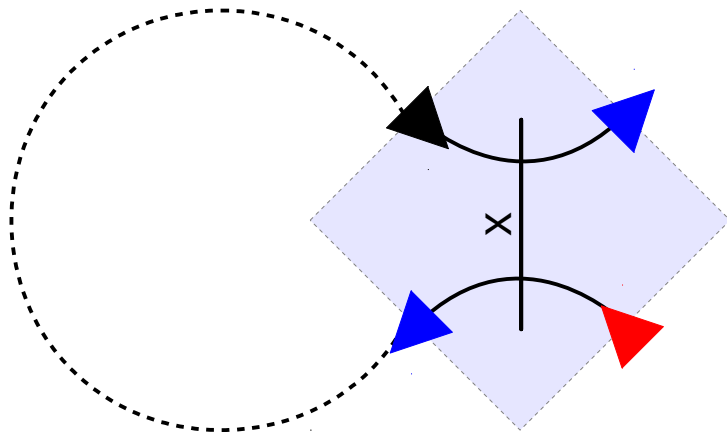
Coloriage des visites (5)



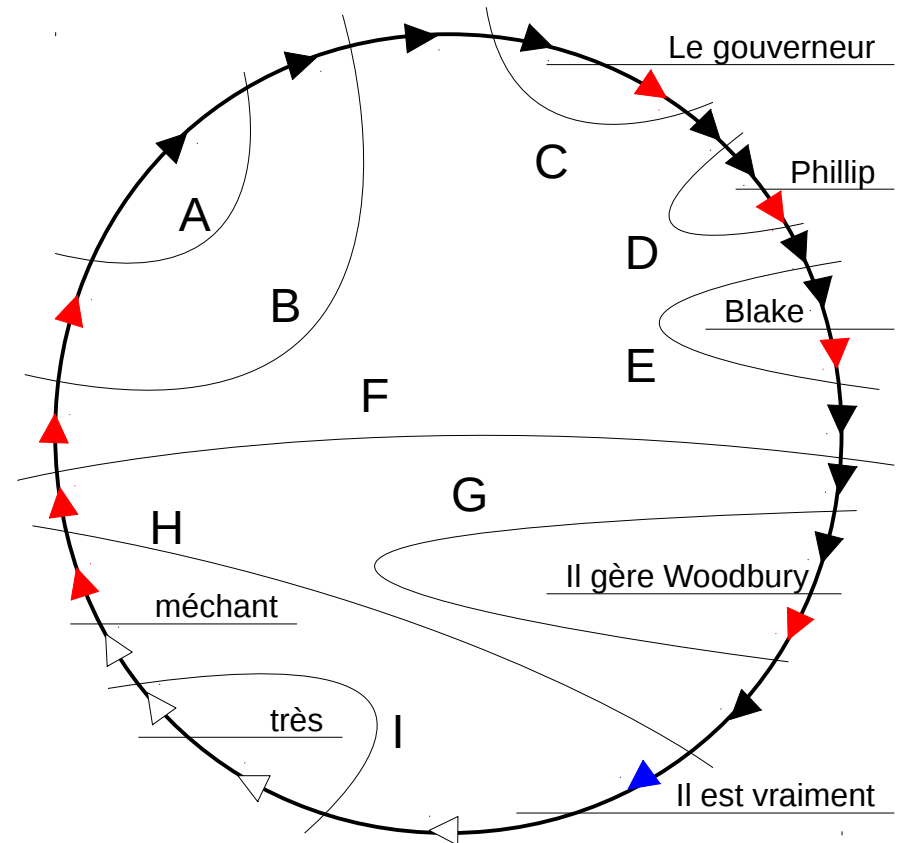
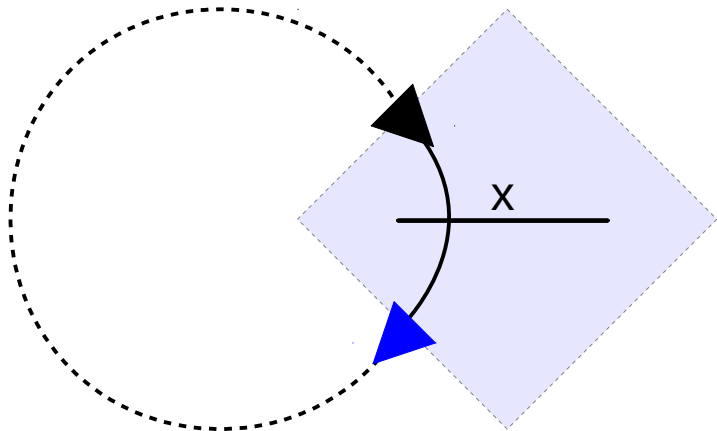
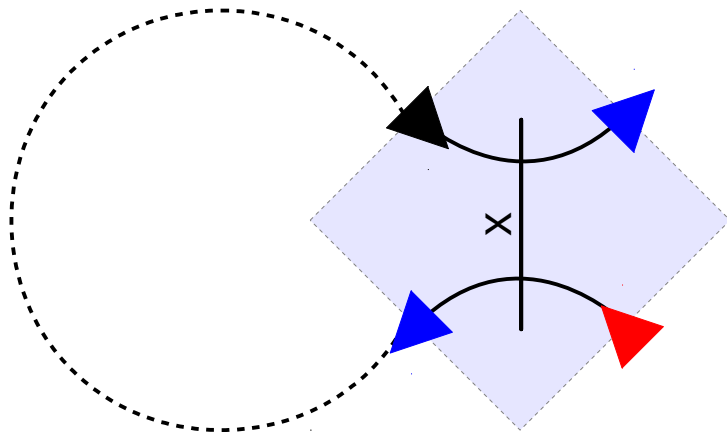
Coloriage des visites (6)



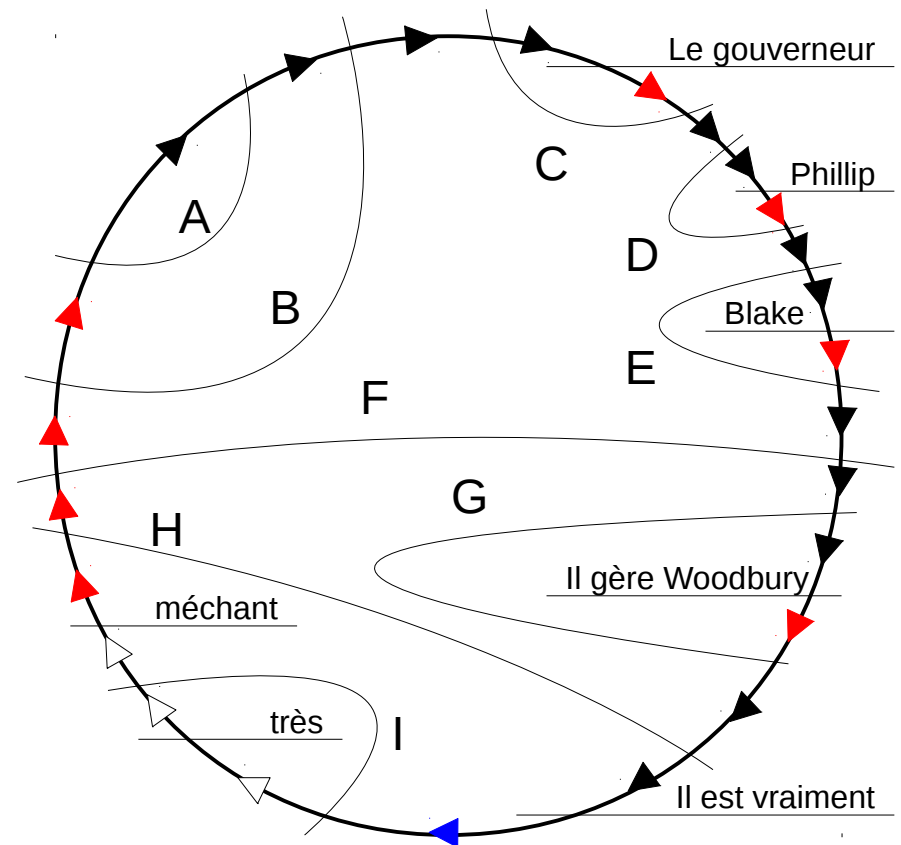
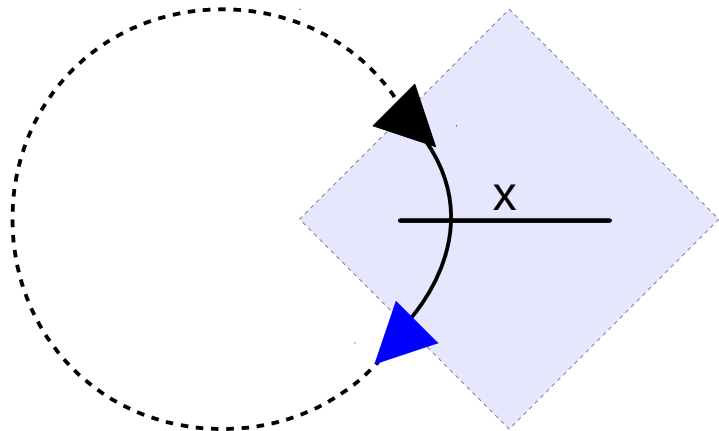
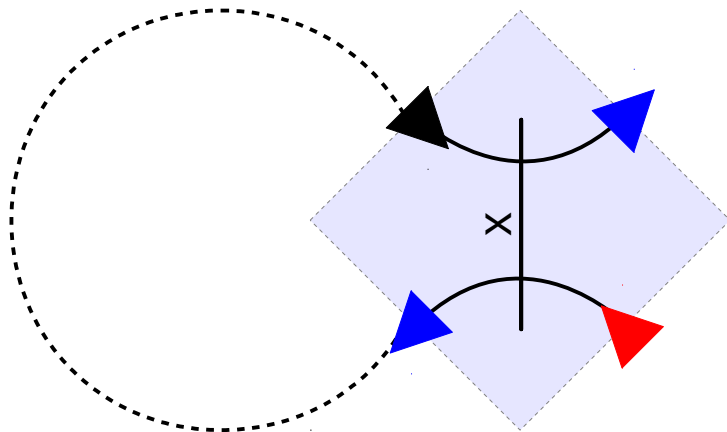
Coloriage des visites (7)



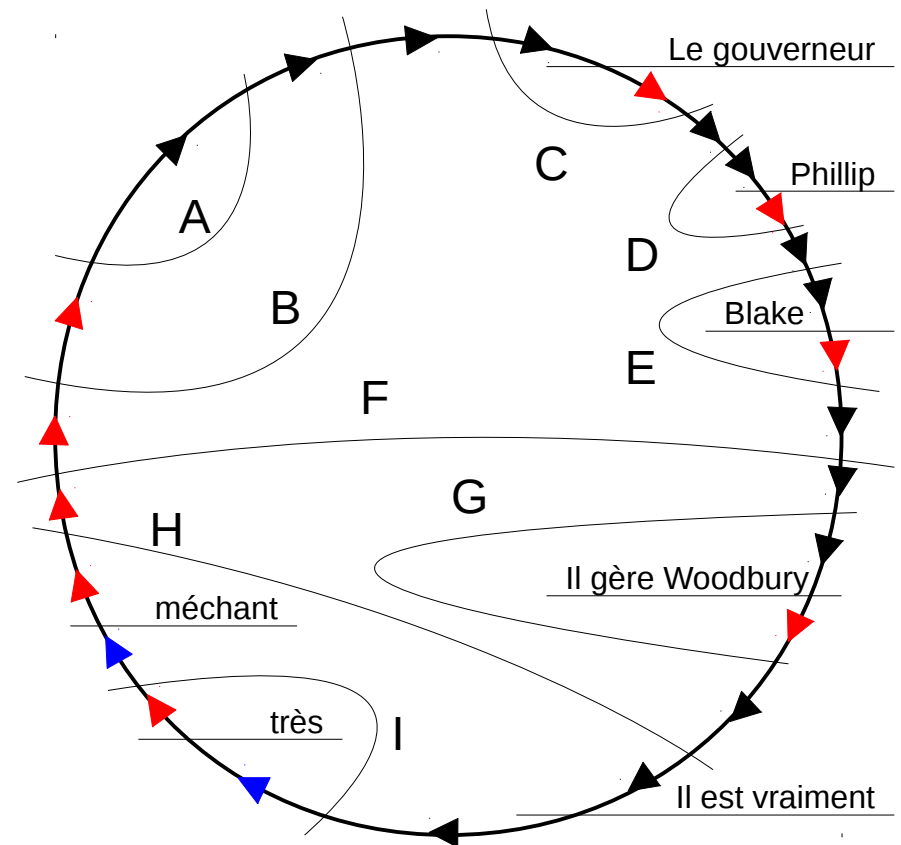
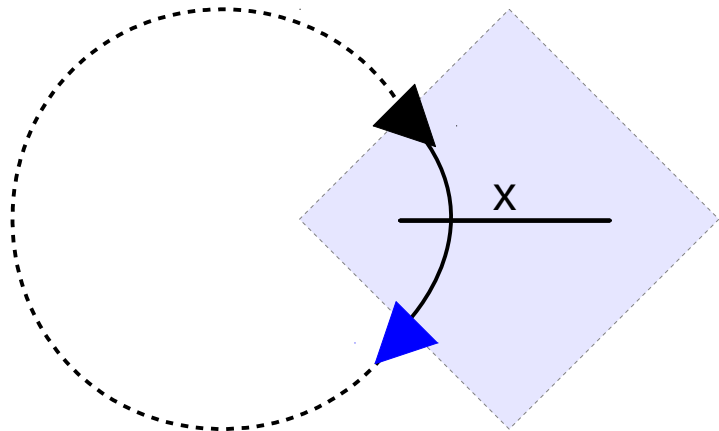
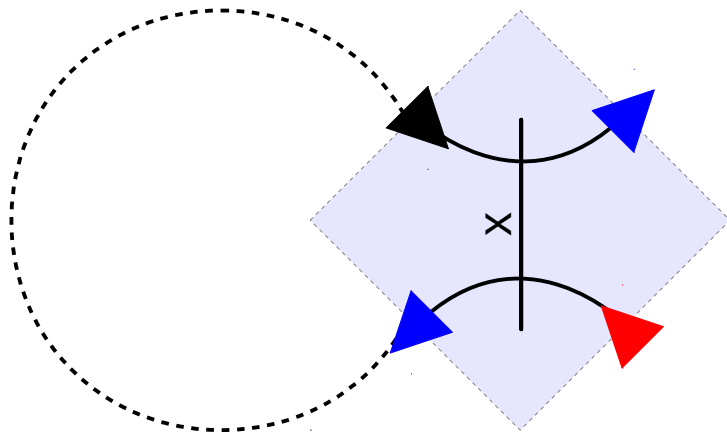
Coloriage des visites (8)



Coloriage des visites (9)

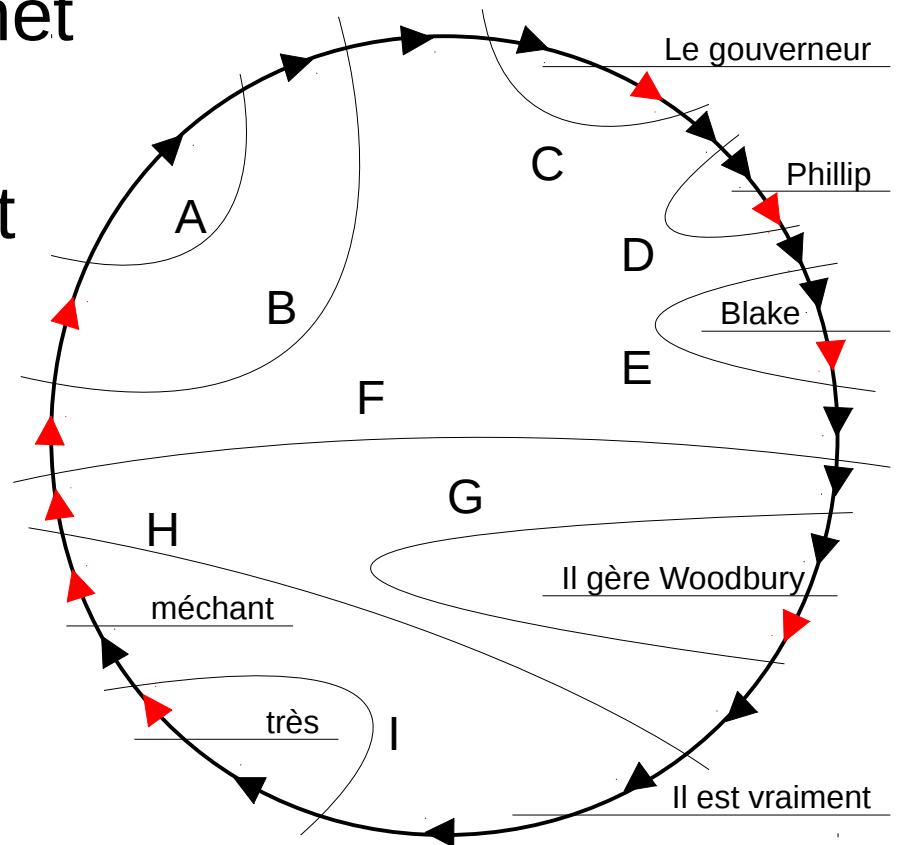
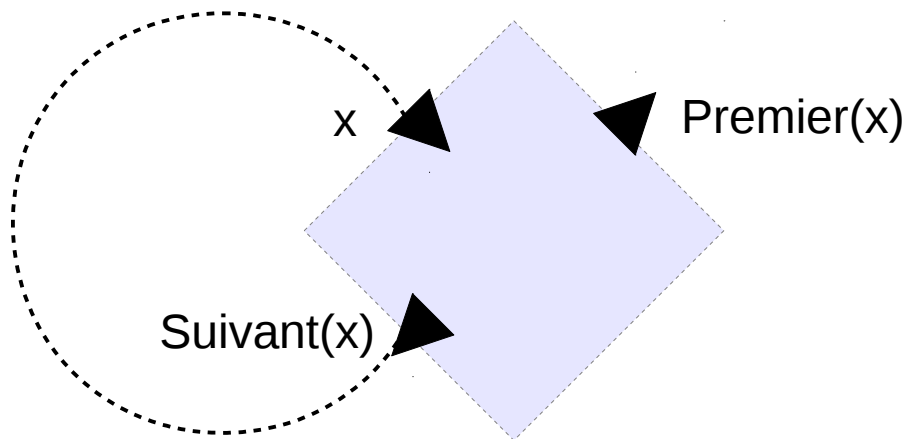


Coloriage des visites (10)



Terminaison

- Chaque visite terminée est un nœud DOM
 - prend la valeur du sommet immédiatement suivant
 - a pour premier et suivant possibles :

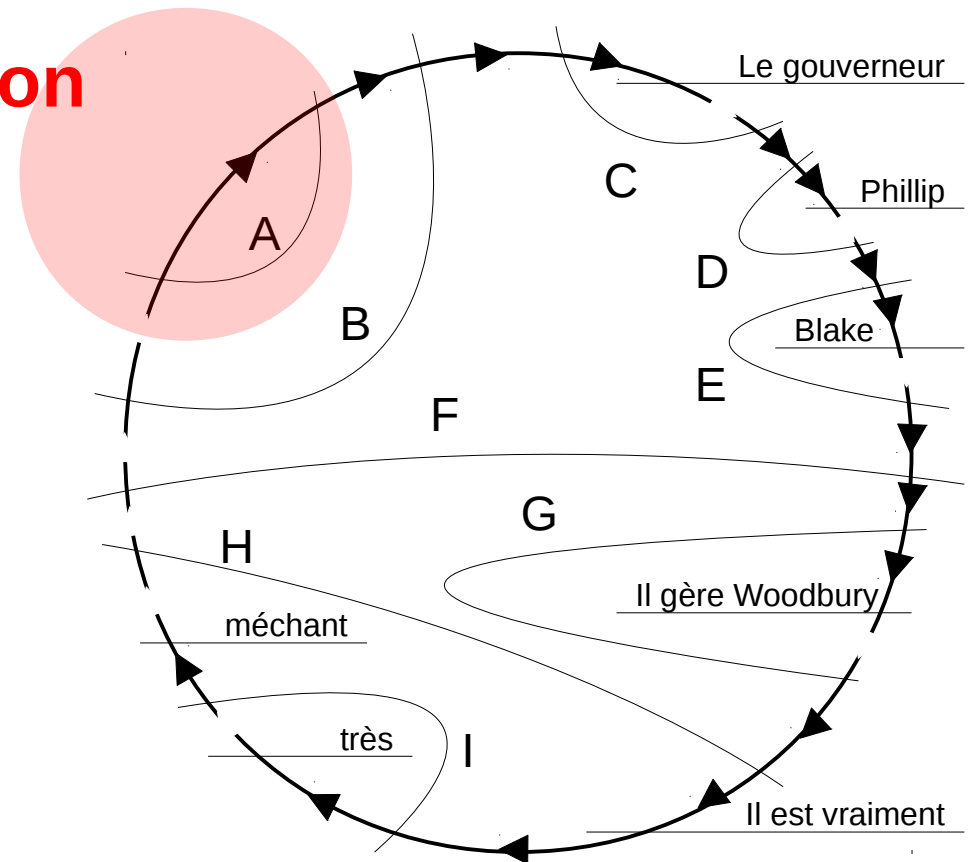


Trois remarques

1. L'initiation « *naïve* » a un effet de bord .
 - Comment le supprimer ?
2. Le parcours est un calcul.
 - Est-il distribuable ou parallélisable ?
3. Notre exemple n'a que des arêtes internes.
 - Comment traiter les arêtes externes ?

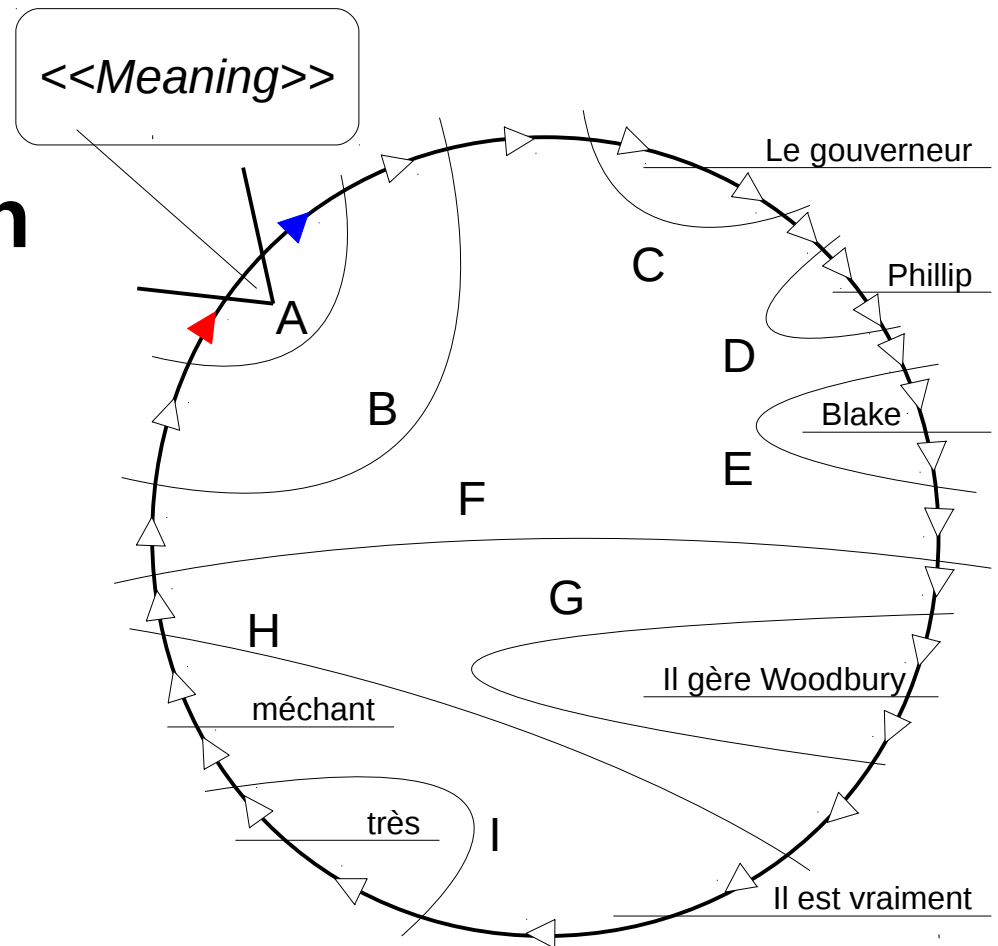
1. Suppression des cycles

- Si nous interprétons le Stop comme une coupure : le coloriage revient à supprimer tous les cycles (→ arbre)
- Demeure la racine
Effet de bord de l'initiation « naïve »
- Il faut :
 - Un équivalent du nœud document
 - Indiquer le choix de de la visite initiale



1. Position et Initiation

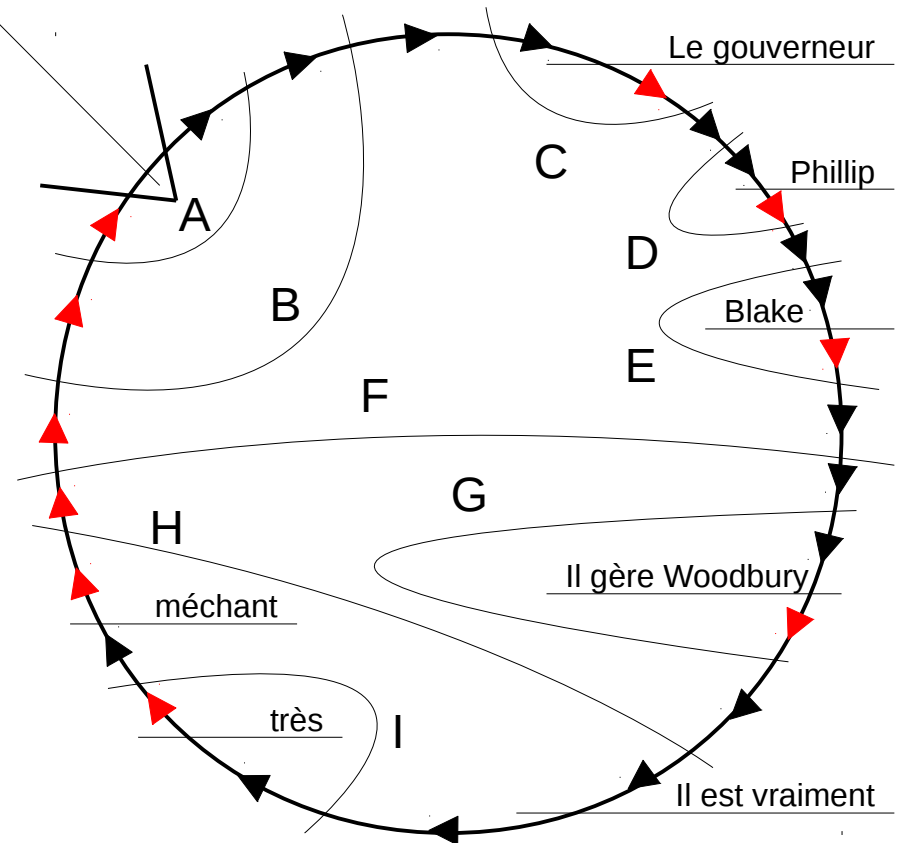
- Nous introduisons un signe : V
- Indique l'initiation du parcours : **la position**
- Équivalent du **nœud document** du DOM : la racine



1. Restitution du fichier initial

```
<personnages>
  <personnage id="42">
    <pseudo>Le gouverneur</pseudo>
    <prenom>Phillip</prenom>
    <nom>Blake</nom>
    <bio>
      <p>Il gère Woodbury</p>
      <p>Il est vraiment
        <strong>très</strong>
        méchant
      </p>
    </bio>
  </personnage>
</personnages>
```

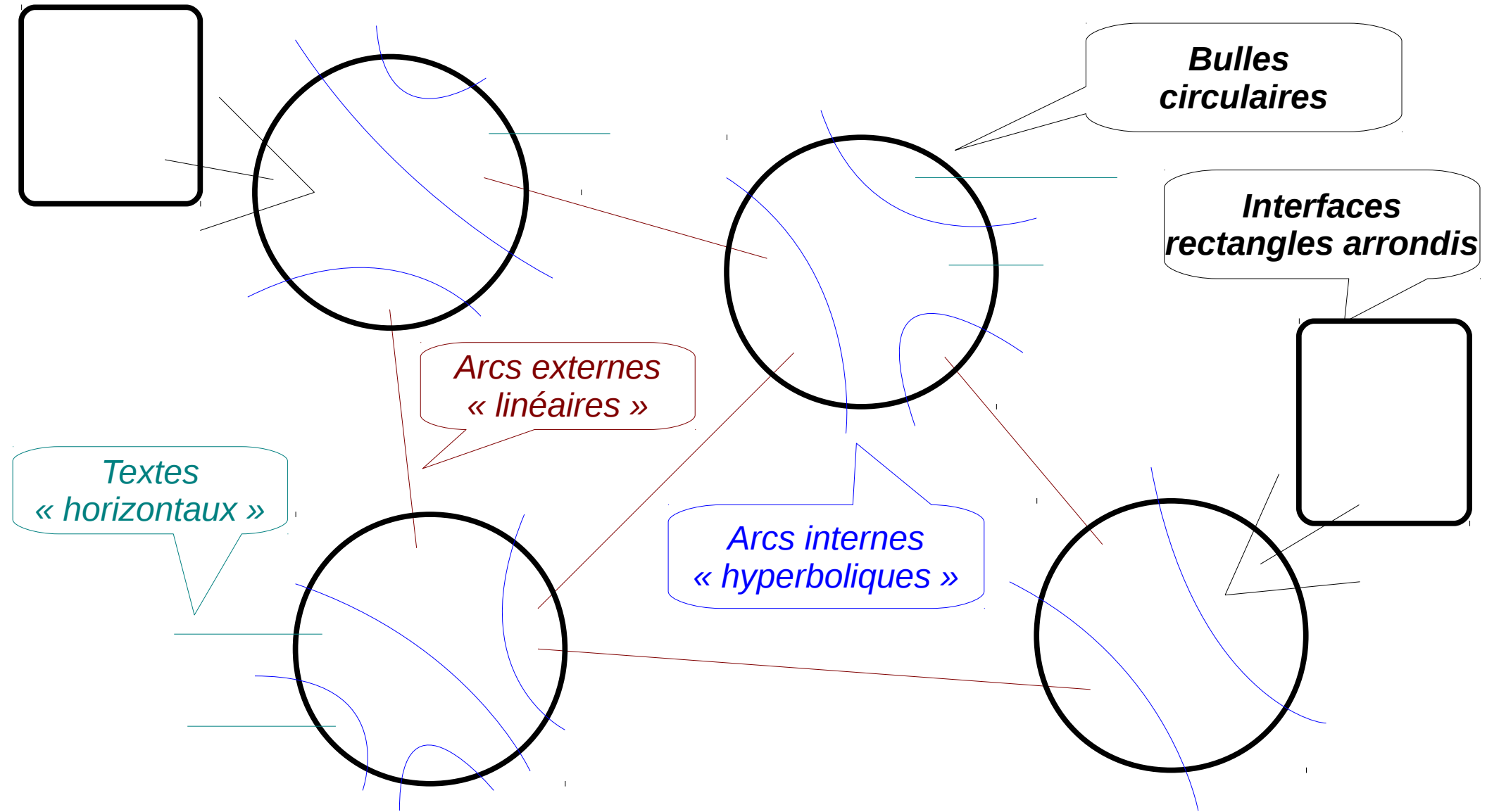
A : personnages
B : personnage id="42"
C : pseudo
D : prenom
E : nom
F : bio
G : p
H : p
I : strong



2. Parcours distribué

- L'obtention du fichier XML associé à une position est un processus réparti.
 - En 2011, nous avons réalisé une maquette où chaque arête externe était associé à une paire de socket TCP, avec succès (Mirza).
- Nous allons maintenant voir comment la vague « blockchain » a impacté notre réflexion et l'usage que nous allons faire de cette capacité de distribution intrinsèque.

3. Que faire des arêtes externes ?



Plan de l'exposé

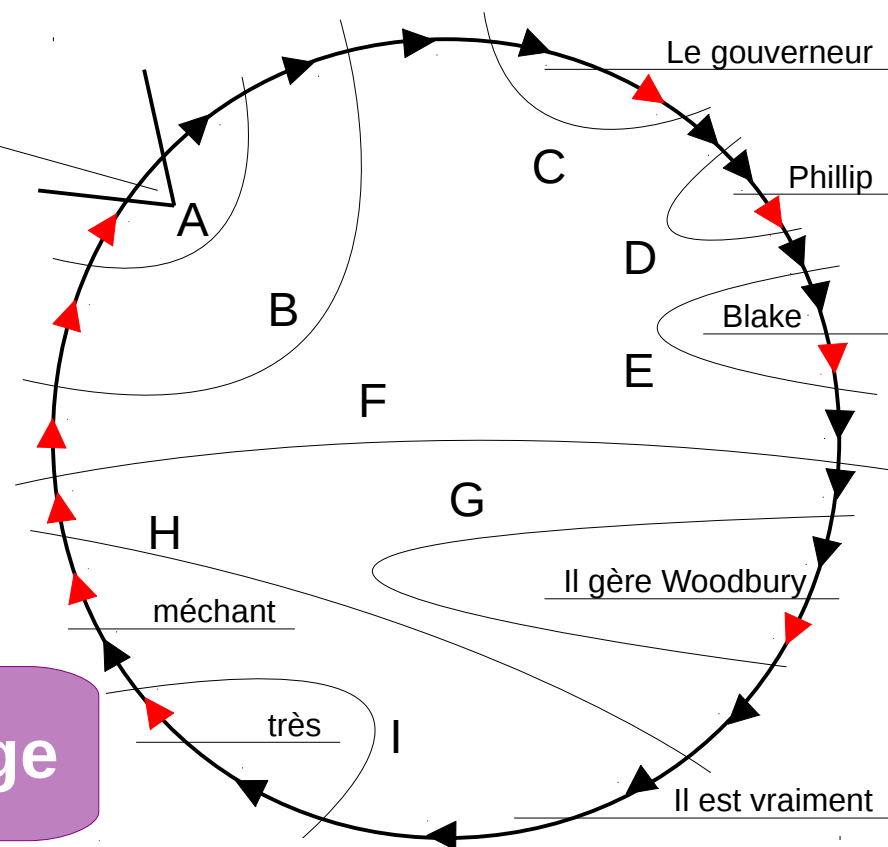
- Créer une topologie calculable
- Traiter les données semi-structurées
- **Intégration du temps**
 - Modèle «Push-Pull»
 - Perspectives et décision
- Conclusion

Modèle push-pull

- Comment traiter une édition directe sur l'interface de visualisation ?

```
<personnages>
  <personnage id="42">
    <pseudo>Le gouverneur</pseudo>
    <prenom>Phillip</prenom>
    <nom>Blake</nom>
    <bio>
      <p>Il gère Woodbury</p>
      <p>Il est vraiment
        <strong>très</strong> trop
        méchant
      </p>
    </bio>
  </personnage>
</personnages>
```

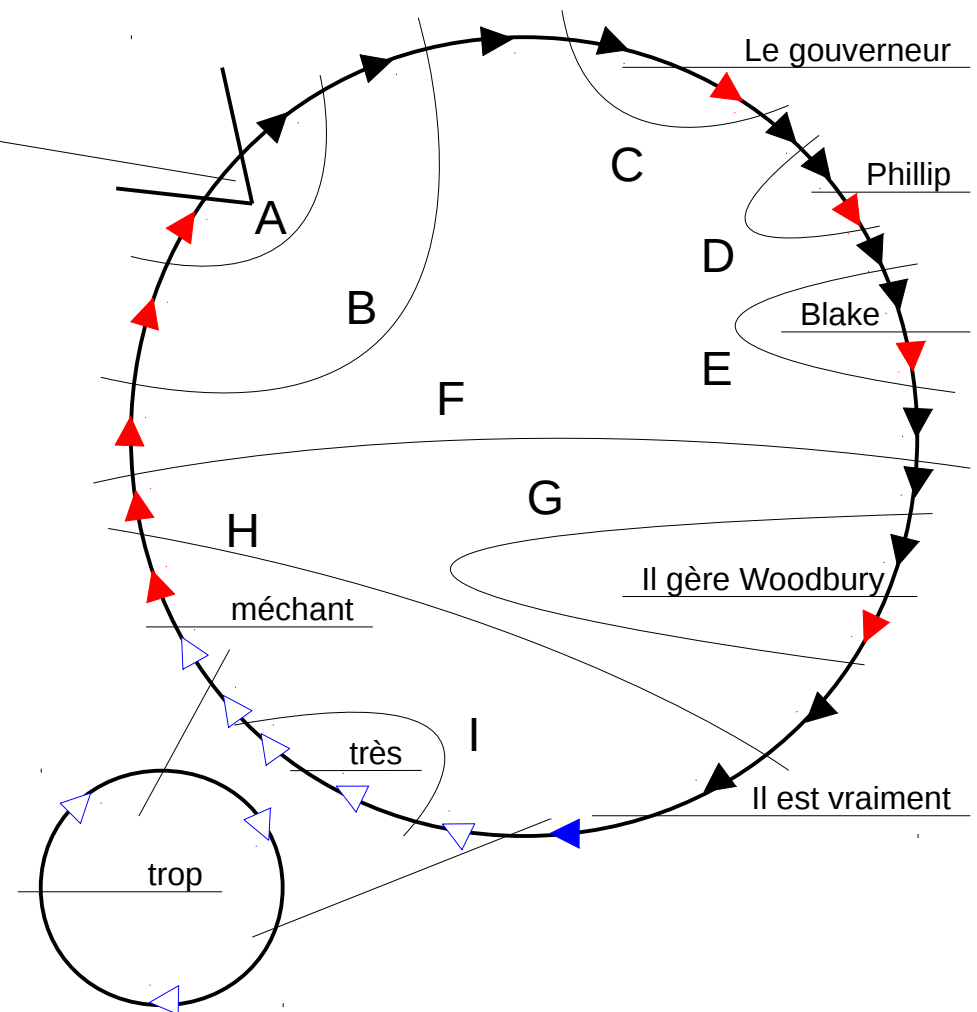
Surcharge



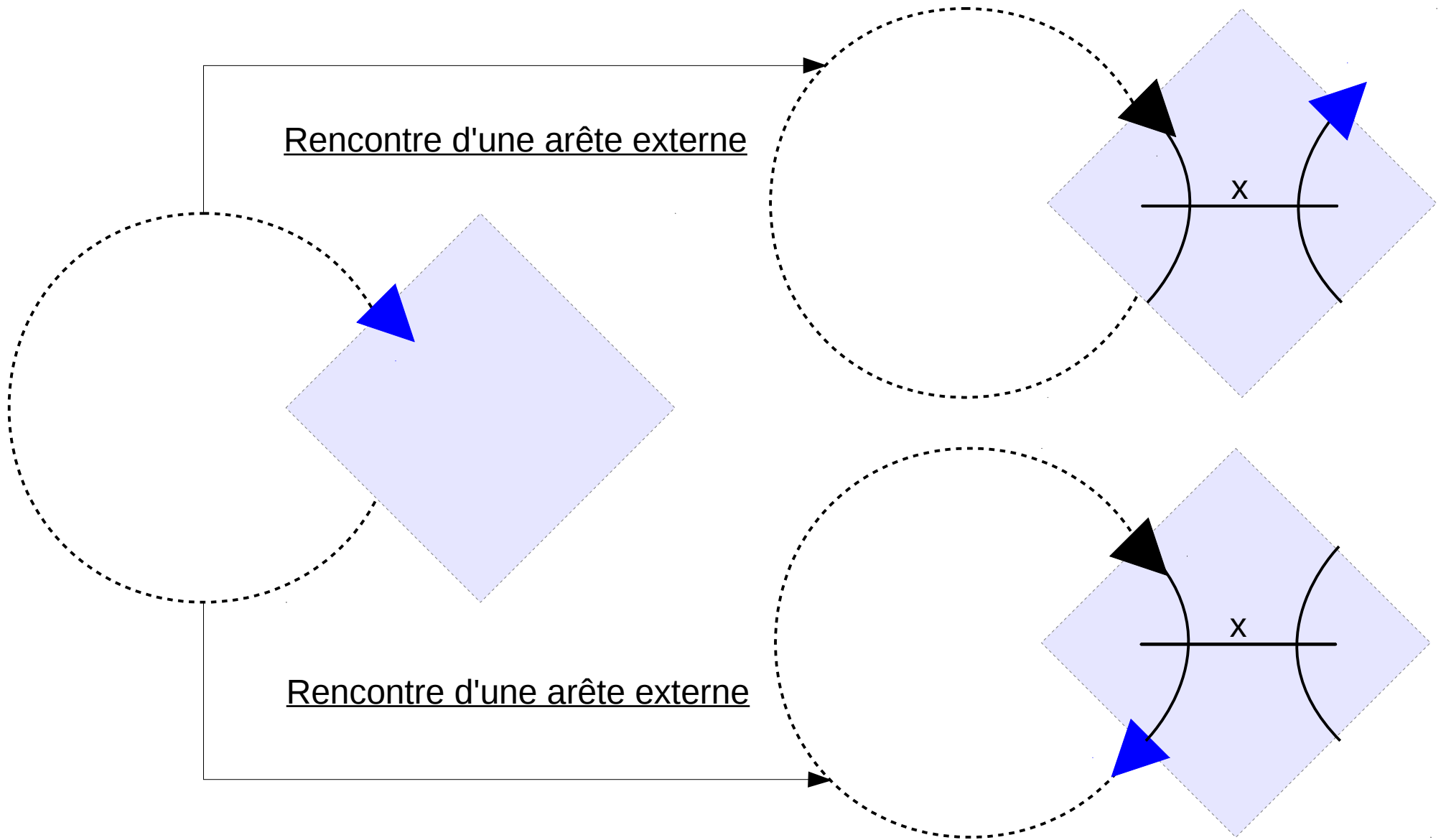
Branchement conditionnel

- Comment traiter une édition directe sur l'interface de visualisation ?

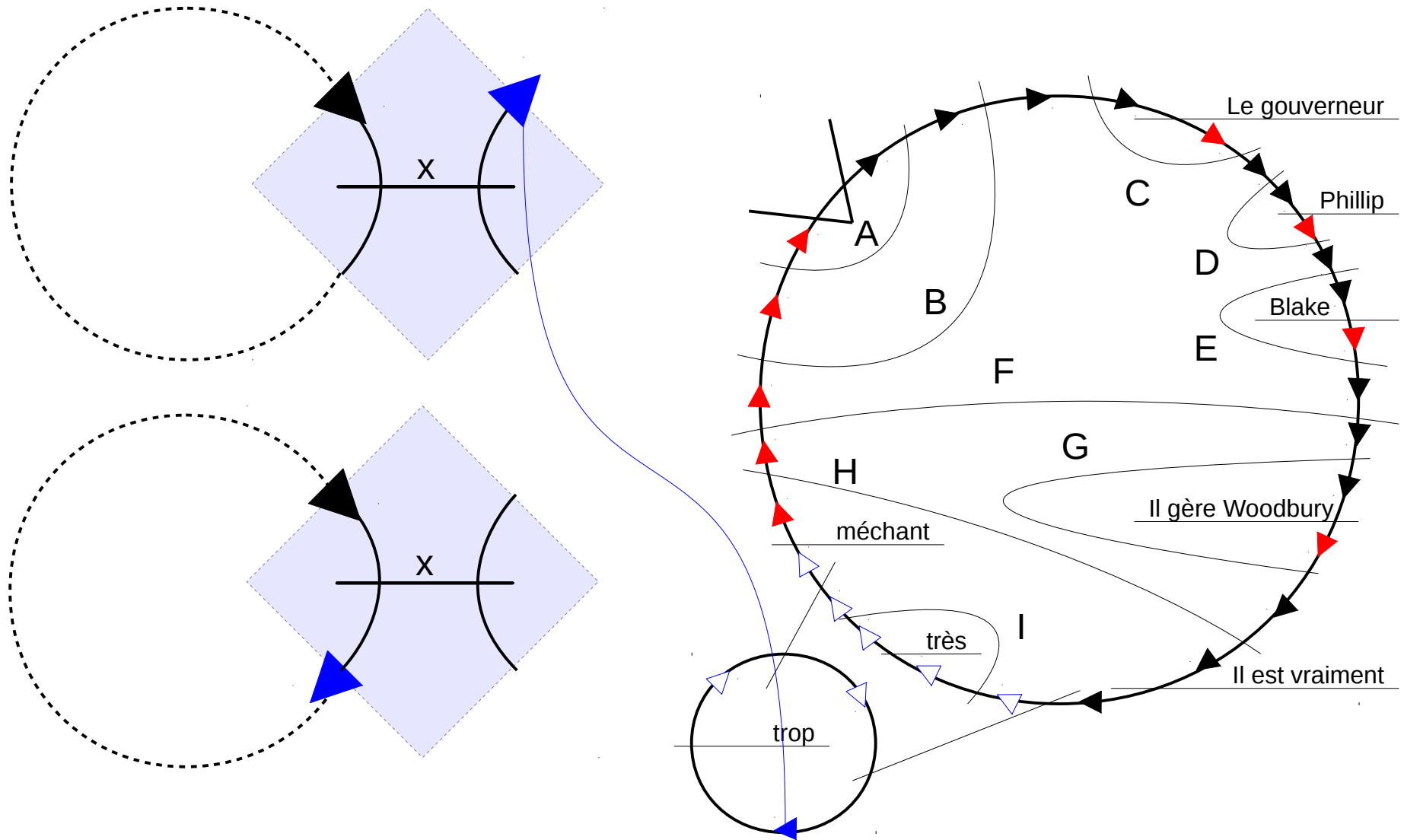
```
<personnages>
  <personnage id="42">
    <pseudo>Le gouverneur</pseudo>
    <prenom>Phillip</prenom>
    <nom>Blake</nom>
    <bio>
      <p>Il gère Woodbury</p>
      <p>Il est vraiment
        <strong>très</strong> trop
        méchant
      </p>
    </bio>
  </personnage>
</personnages>
```



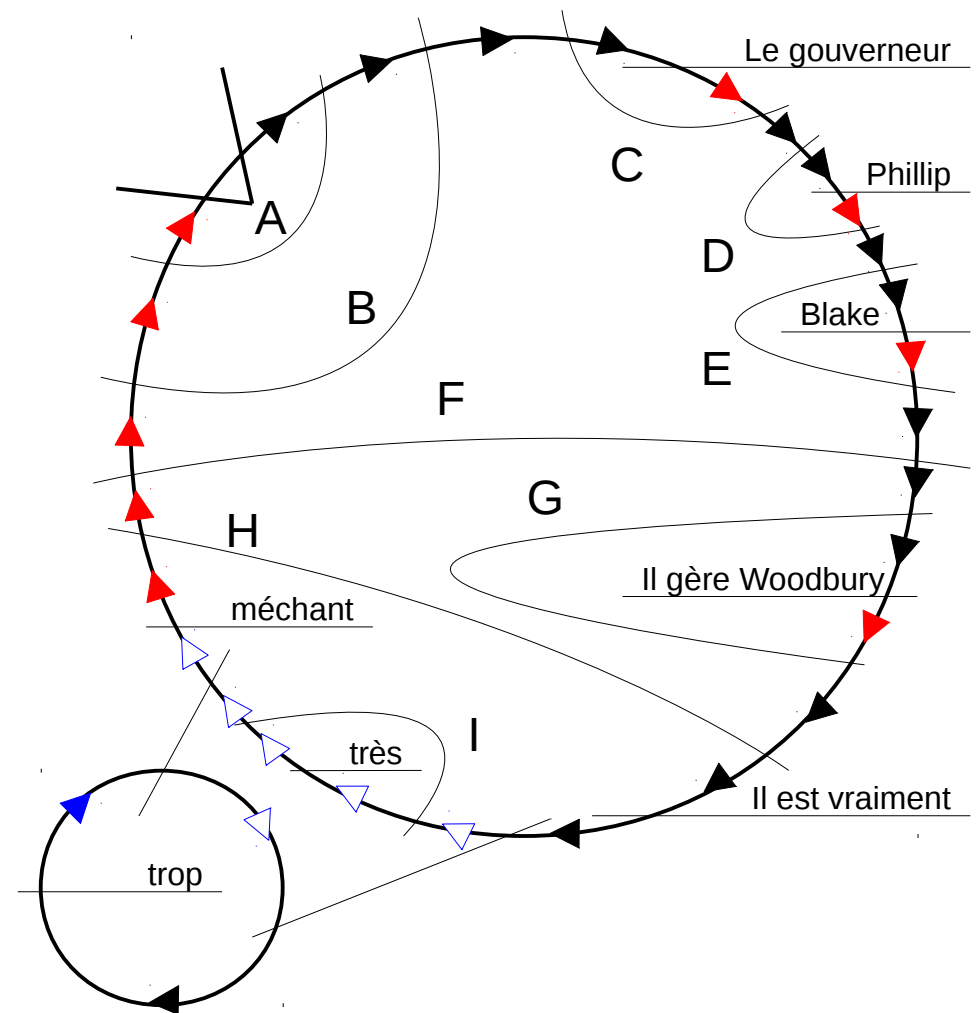
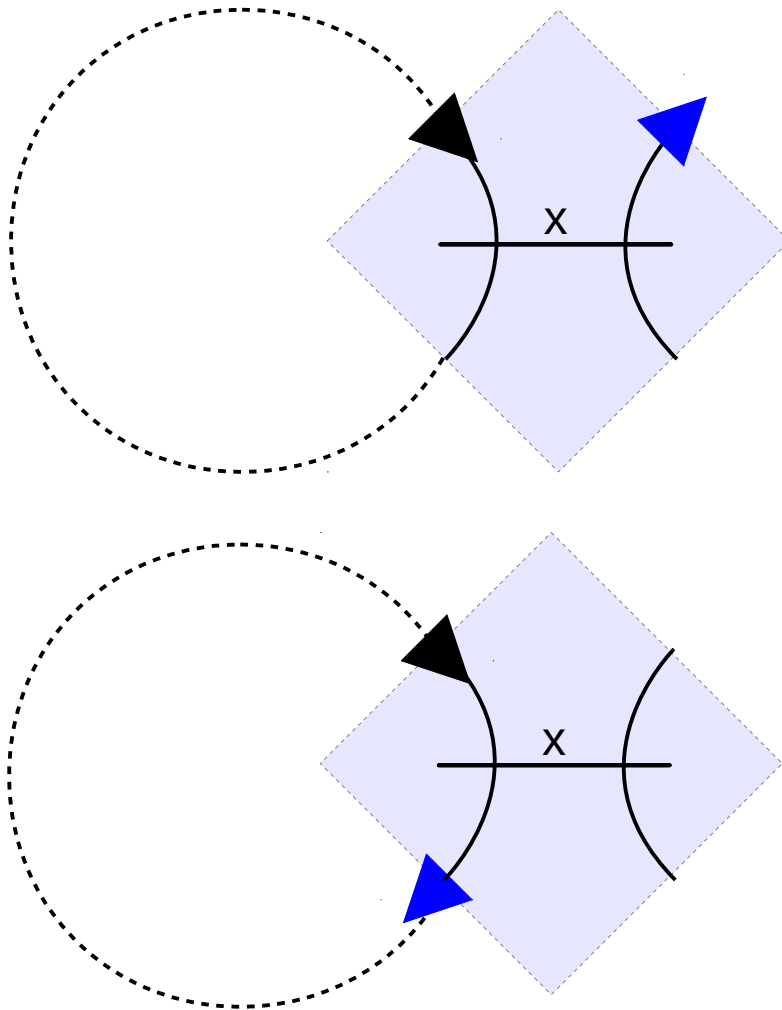
Décider les arêtes externes



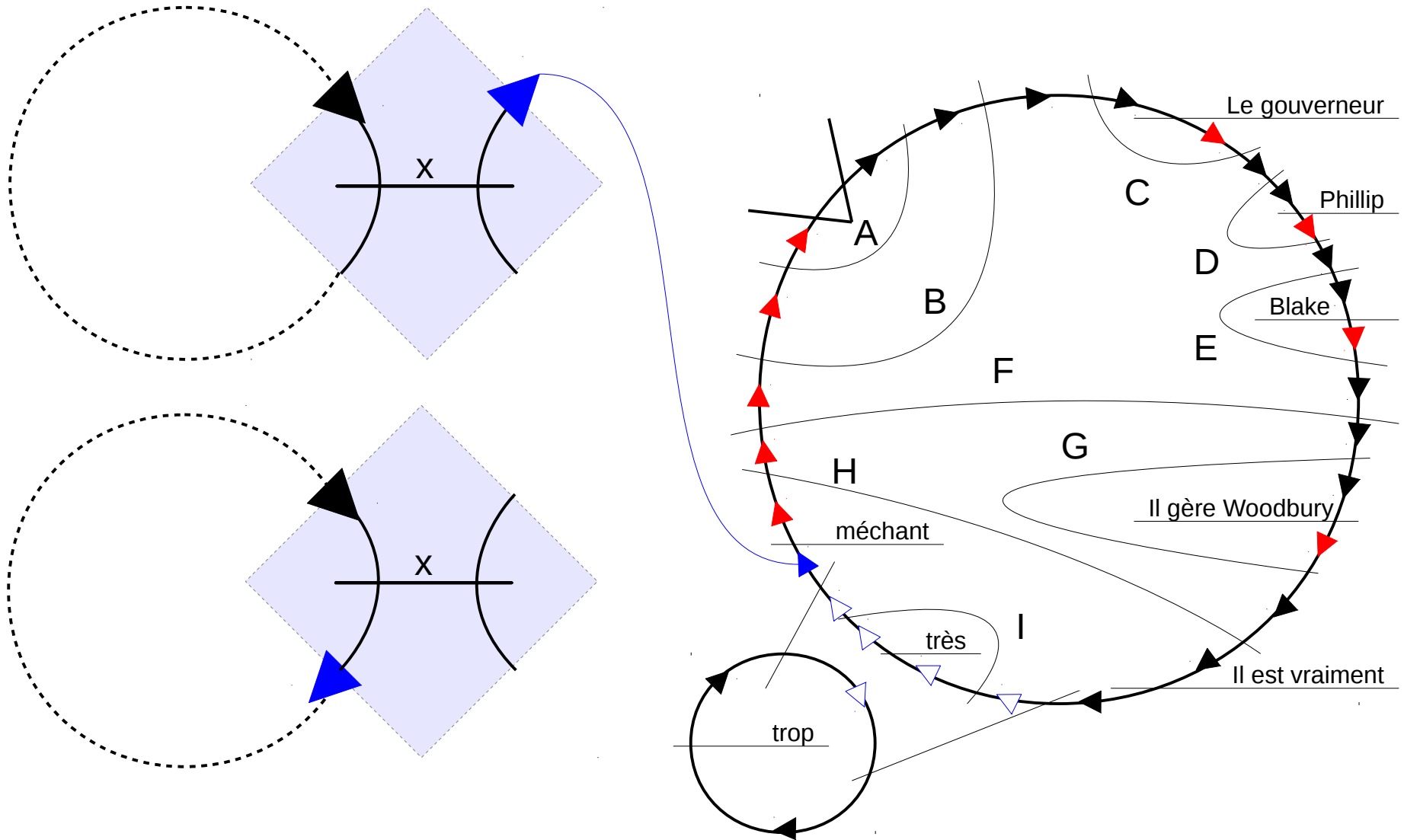
Parcours conditionnel (1)



Parcours conditionnel (2)

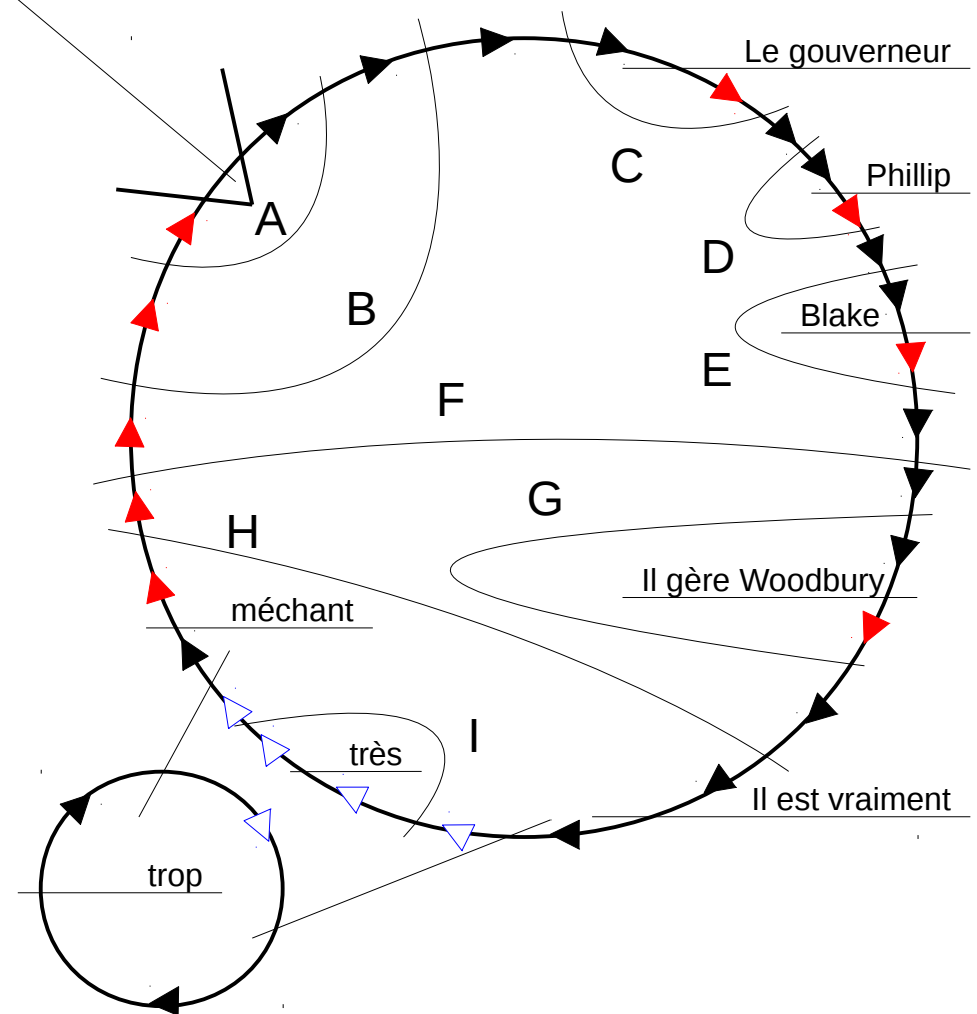


Parcours conditionnel (3)



Terminaison alternative

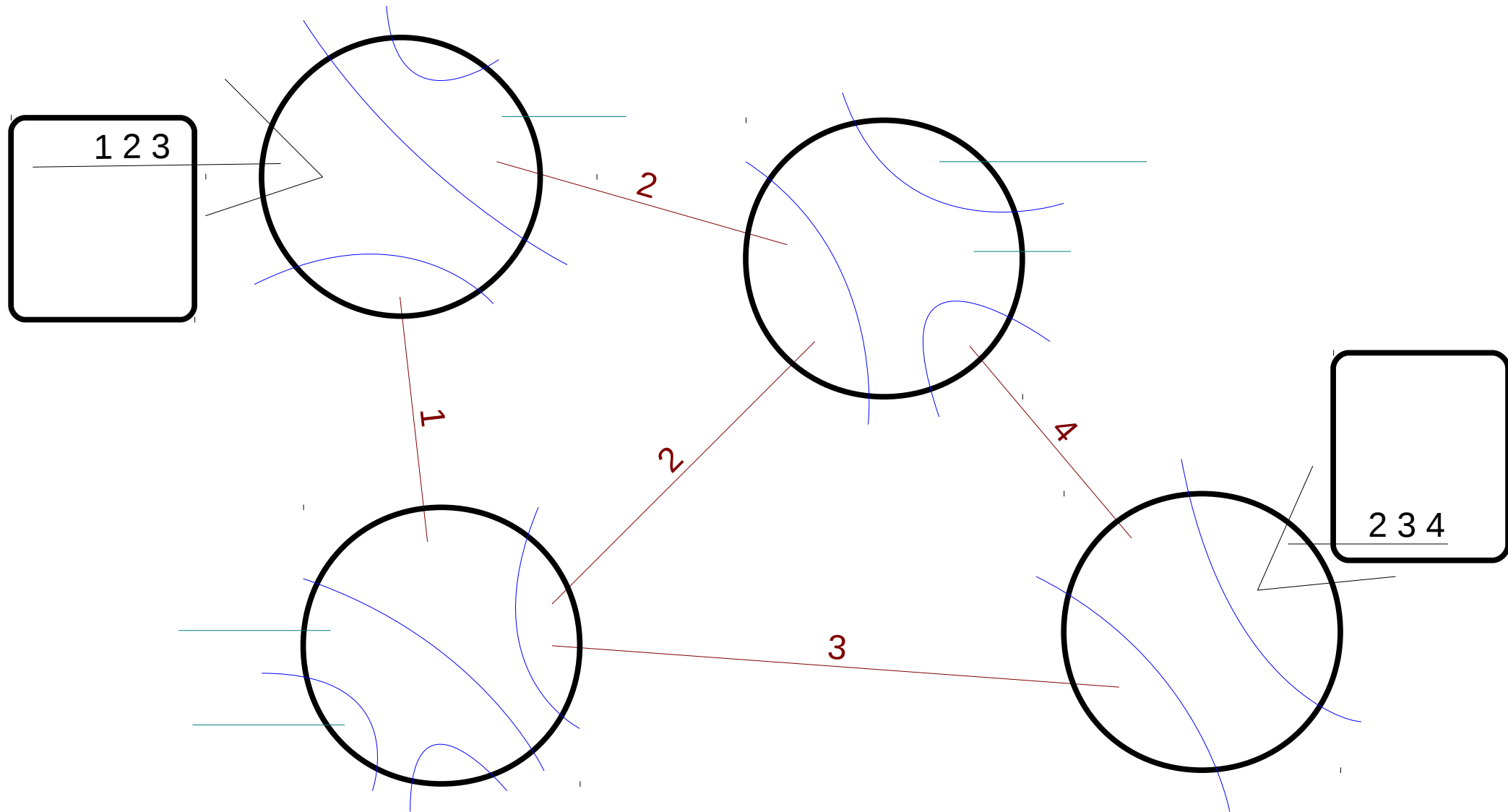
```
<personnages>
  <personnage id="42">
    <pseudo>Le gouverneur</pseudo>
    <prenom>Phillip</prenom>
    <nom>Blake</nom>
    <bio>
      <p>Il gère Woodbury</p>
      </p>
      <p>Il est vraiment trop méchant</p>
      </p>
    </bio>
  </personnage>
</personnages>
```



Perspective de Winskel (TED)

- Les structures d'événements de Winskel peuvent être utilisées pour définir les perspectives :
 - Une perspective est un ensemble d'arêtes externes
 - Stratégie utilisée dans le prototype TED de 2015.
 - L'appartenance est le critère de décision
 - L'intersection de perspectives est une opération interne consistante.
 - La dépendance définit une relation d'ordre partiel sur les arêtes externes :
 - compression des perspectives (seuls les éléments de surface peuvent être stockés)
 - La linéarisation répartie de l'ordre partiel est l'analogue de l'algorithme de consensus de la blockchain (Octobre 2015 – Bon sang mais c'est... Bien sûr)

Linéarisation du diagramme normal



Parcours programmables (CARGO)

SUIVI SUR
CARTES

REGISTRE
DES ACTES

SUIVI SUR
TABLEAUX

REGISTRE
DES ACTES

UNE SEULE
PLATEFORME

GESTION DE
PROCESSUS

REGISTRE
DES ACTES

ANALYSE
DE CODE

REGISTRE
DES ACTES

XSL

XSL

XSL

XSL

Plan de l'exposé

- Créer une topologie calculable
- Traiter les données semi-structurées
- Intégration du temps
- **Conclusion**
 - Vues et dimensions structurantes
 - Perspectives de travaux futurs

Vue = Dimension structurante

Lieux

L'hôpital

Atlanta

Le camp des survivants

Le C.D.C.

L'autoroute

La ferme des Greene

La prison

Woodbury

Le Terminus

L'église Sainte Sarah

Le Grady Memorial Hospital d'Atlanta

Alexandria Safe Zone

La colline

Personnages

Aaron

Jessie Anderson

Philip "le Gouverneur" Blake

Tara Chambler

Daryl Dixon

Rosita Espinosa

le sergent Abraham Ford

Gareth

Beth Greene

Hershel Greene

Maggie Greene

Rick Grimes

Carl Grimes

Morgan Jones

Michonne

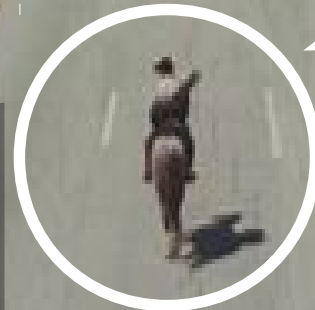
Deanna Monroe

Spencer Monroe

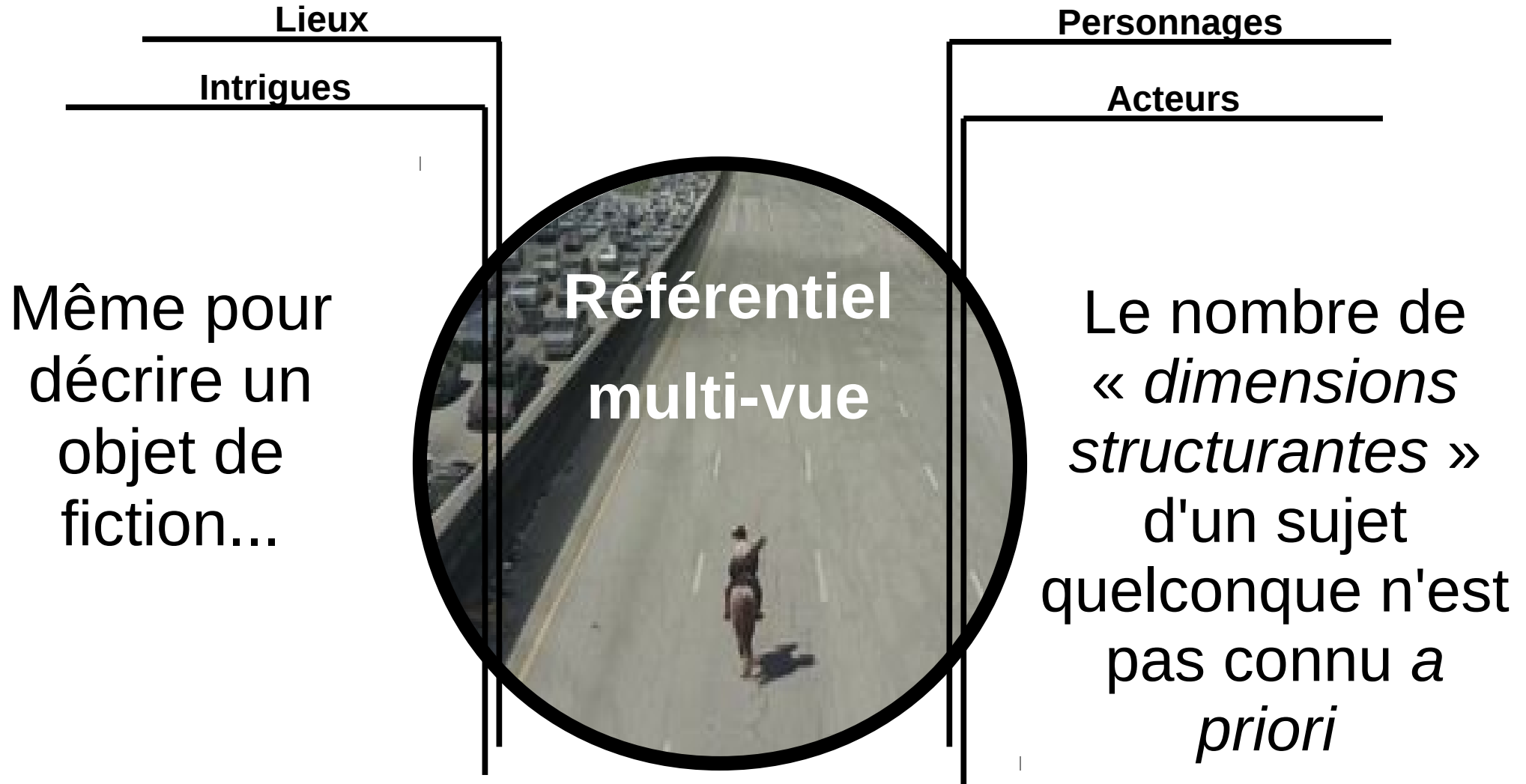
Negan

Carol Peletier

La première saison de « *Walking Dead* » introduit le personnage **Rick Grimes**, qui se réveille à l'hôpital après un long coma de plusieurs mois. Il découvre avec effarement que la population entière, ravagée par une épidémie d'origine inconnue, est envahie par les morts-vivants. Parti sur les traces de sa femme et de son fils, Rick arrive à **Atlanta** où, avec un groupe de rescapés, il va devoir apprendre à survivre.

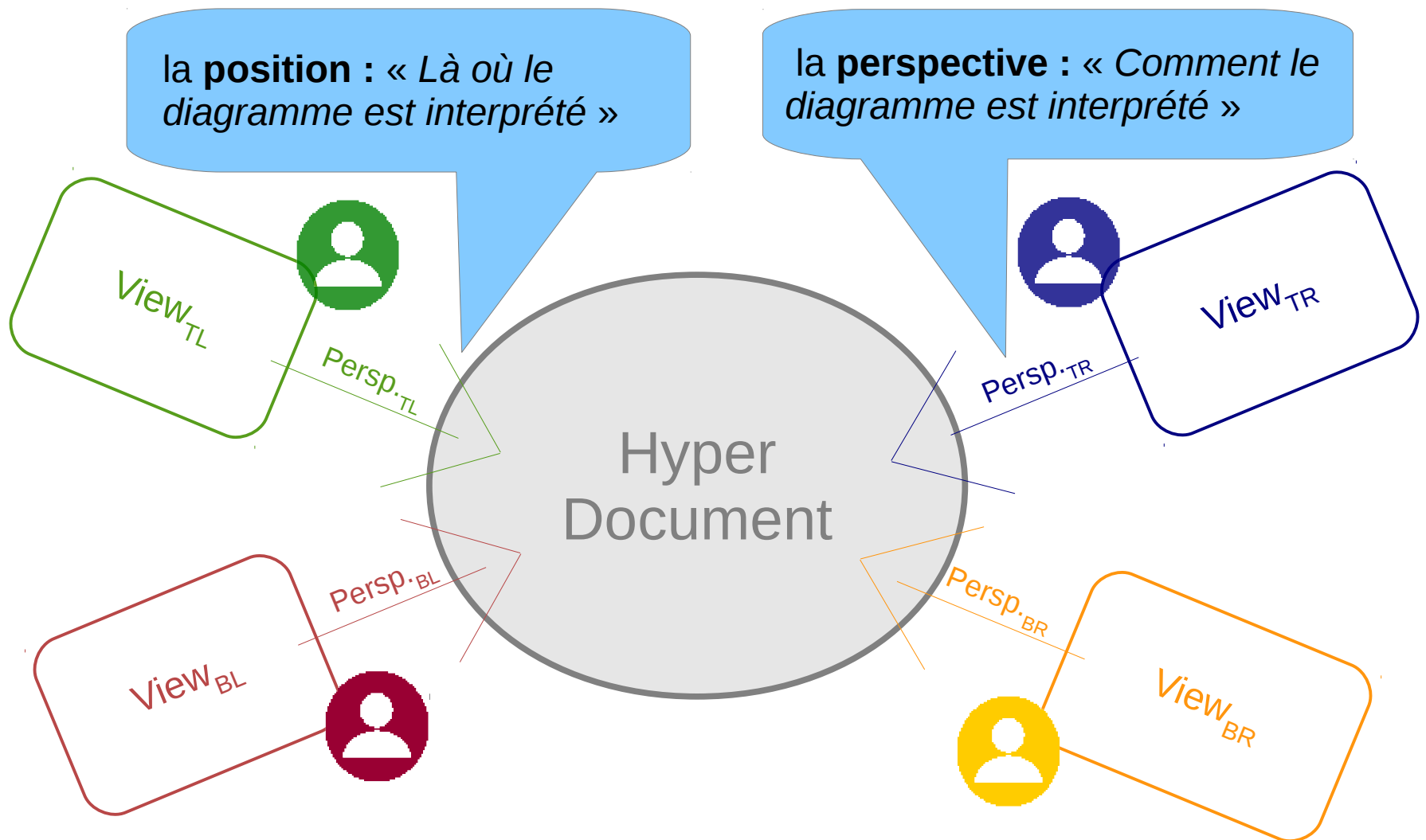


Dans un contexte social

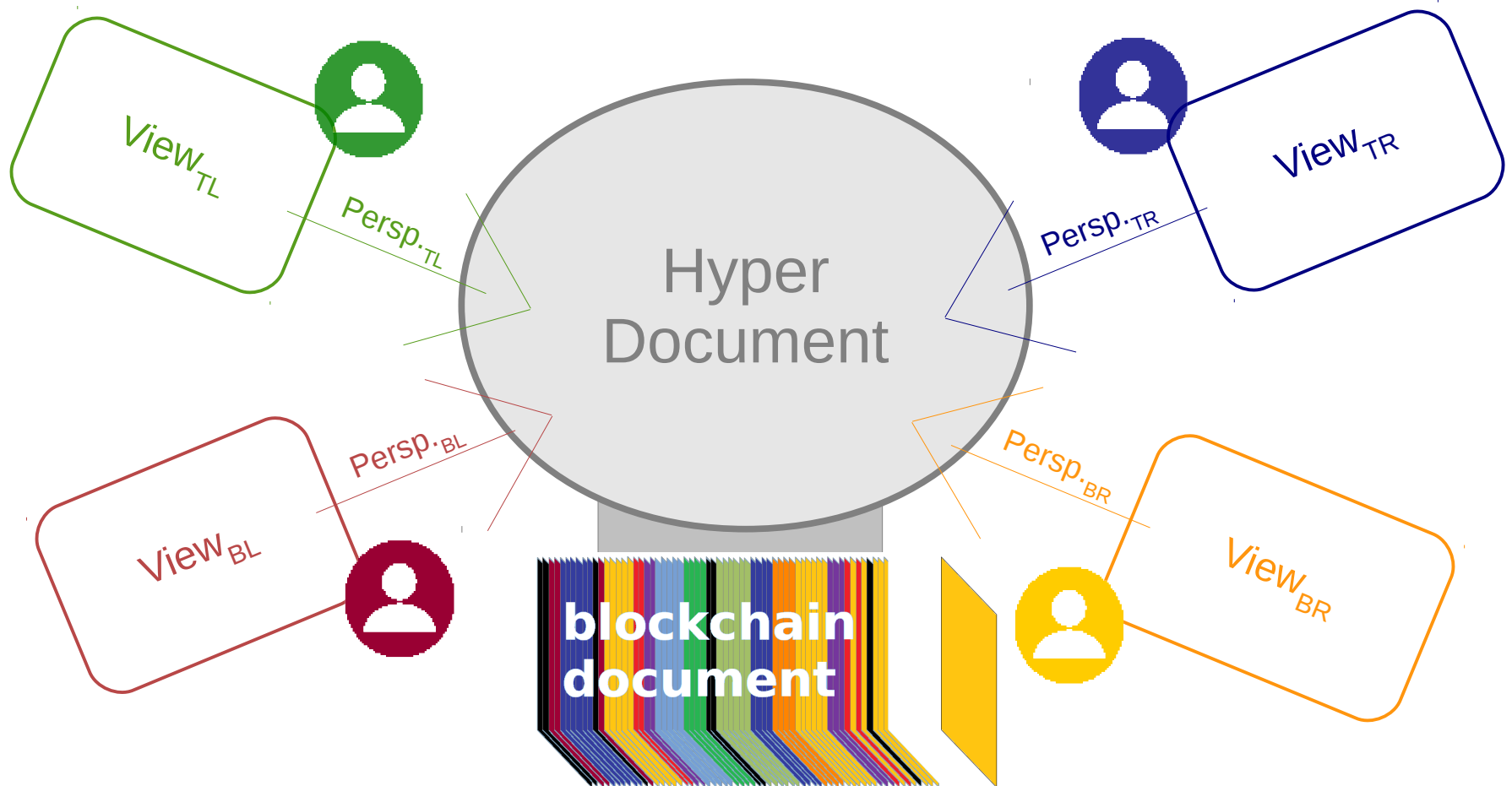


Modélisation des vues

Dans notre modèle d'hyper-édition exprimé en formulation modale, une vue est l'association de deux dimensions :



Hyper-édition



L'hyper-document est l'espace-temps de navigation dans toute l'information du système.

Capacités de formulation modale

- La formulation permet de répondre à tous les enjeux de la représentation de la **donnée socialisée** :
 - Représenter les graphes de façon calculable
 - Traiter les données semi-structurées
 - Représenter les graphes de façon efficiente
 - Intégrer le temps dans la représentation
- En dans ces capacités, la **dualité**, une opération sans fondement solide en théorie classique, devient la pierre angulaire de toute l'approche :
 - Permet de lier la topologie du dessin à la représentation informatique
 - Permet de passer de la forme textuelle à la forme graphique de l'XML
 - Permet de faire toutes les transformations de manière massivement parallélisable
 - Permet de représenter le temps et l'espace comme des dimensions duales

Exigence SDI couvertes

- Où le temps est une dimension structurante
 - L'hyper-document est un diagramme spatio-temporel
 - La dualité impose que ce diagramme soit un « *collage temporel* » de termes, que nous avons associé au concept du « *Tetris* ».
- Où les normes du W3C sont applicables
 - Les normes applicables pour toutes les écritures de termes sont les normes du W3C.
 - Notre support de persistance est une liste de lignes, où chaque ligne est homogène à un terme XML et où un calcul de type hash est réalisé sur tous les antécédents de chaque terme.
- Où toute action sur le SDI est un acte à valeur juridique
 - Notre modèle Tetris est une extension de la blockchain où le caractère décentralisé du calcul de l'irréversibilité du temps est étendu à la structure de la donnée – la dualité nous empêchant « *de faire les choses à moitié* ».
- Où chaque acteur peut naviguer à volonté dans toutes les dimensions structurantes du SDI
 - La formulation modale, par la richesse de ces formes de représentation (hyperbolique, normale, ...) permet de créer des interfaces variées et ergonomiques pour toutes complexités et toutes configurations.

Exigence SDI à couvrir

- Où le nombre de serveurs et d'acteurs n'est pas borné a priori
→ **Vues internes** : le symbole de vue et sa sémantique n'ont été que survolé dans cette présentation, leur sémantique peut s'interpréter comme celle d'une API permettant de les utiliser de manière standard comme interface entre systèmes, qu'ils soient pairs ou tiers.
- Où chaque acteur peut naviguer à volonté dans toutes les dimensions structurantes du SDI
→ **Gestion des perspectives** : les perspectives sont des structures d'événements de Winskel avec des possibilités d'utilisations de l'intersection comme aide à la navigation. L'ensemble permet de définir un système de navigation dans toutes les dimensions du système avec l'analyse du réseau social formé autour du point de vue de chaque acteur.
- Où le nombre de dimensions structurantes n'est pas fixé a priori
→ **Parcours interactifs programmables** : les visiteurs produisent des représentations semi-structurées en parcourant les diagrammes. Ces parcours peuvent être paramétrés par des termes, à la manière de XSL et ces parcours peuvent contenir des termes de réécriture (« termes de preuve ») pouvant être inséré dans le diagramme espace-temps (ou hyper-document)