

Comparando Feature Notations

—

Luciano Marchezan

Roteiro

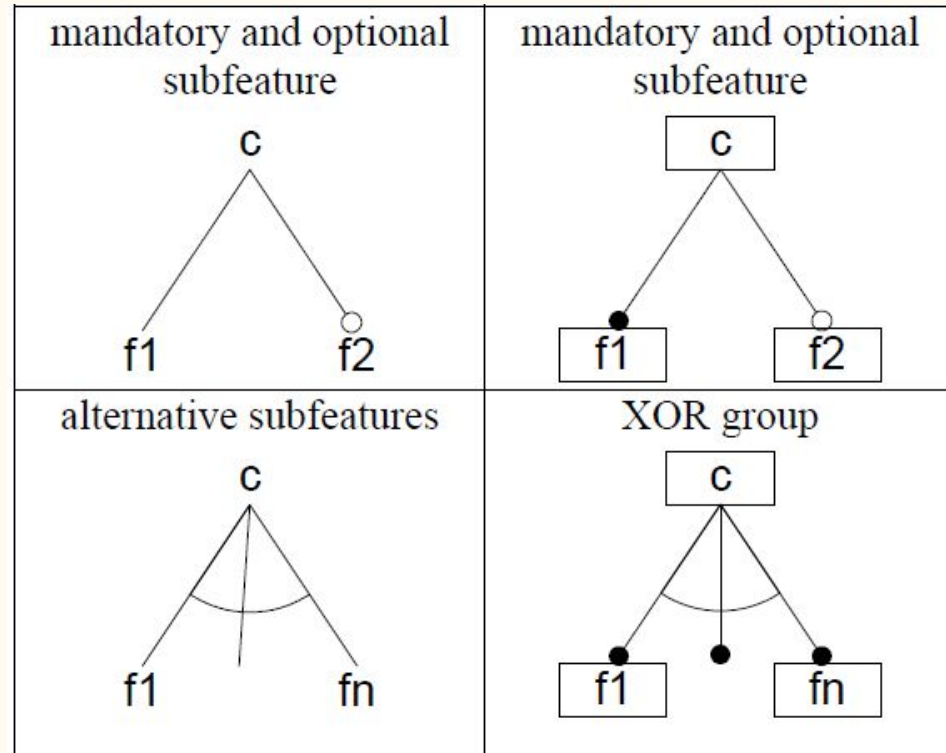
- Comparação geral
- FODA vs GP
- GP vs CB
- GP vs CBS
- CB vs CBS
- GP vs FeatuRSEB

FODA	General Programming	Cardinality-based	FeatureRSEB	GBS Notation
<i>Mandatory Features</i>	<i>Mandatory Features</i>	<i>Mandatory Features</i>	<i>Normal Feature</i>	<i>Mandatory Feature</i>
<i>Optional Features</i>	<i>Optional Features</i>	<i>Optional Features</i>	<i>Static Binding Features</i>	<i>Optional Feature</i>
<i>Alternative Features</i>	Grupo com <i>XOR-features</i>	Grupo com cardinalidade <1-1>	<i>Static Binding Features</i>	<i>XOR specialization</i>
-	Grupo com <i>OR-features</i>	Grupo com cardinalidade <1-k>	<i>Dynamic Binding</i>	<i>OR specialization</i>
-	Grupo com <i>optional XOR-features</i>	Grupo com cardinalidade <0-1>	-	-
-	Grupo com <i>optional OR-features</i>	Grupo com cardinalidade <0-k>	-	-
-	-	-	-	<i>External Features</i>

FODA vs GP

—

FODA vs GP



FODA vs GP

- FODA não permite a criação desse grupo.

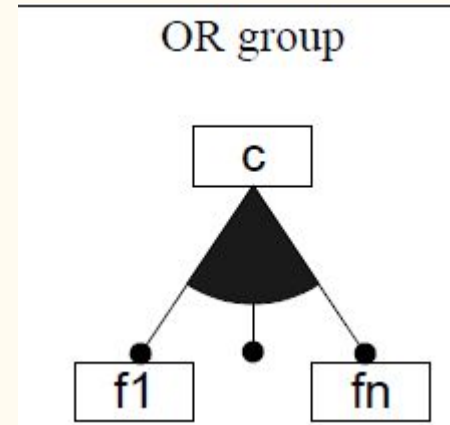
Opções:

- *Features* alternativas
 - Não permite selecionar **mais de uma**.
- Grupo com *features* opcionais
 - Permite **não** selecionar uma *feature*
- Grupo com *features* opcionais e uma *feature* obrigatória
 - Limita à **sempre** selecionar a *feature* obrigatória

Possível solução: Combinar opções anteriores com regras de composição. (Gambiarra?)

- *Or-Features*

- Grupo de *features* alternativas onde 1 *feature* **deve** ser selecionada e várias *features* **podem** ser selecionadas,



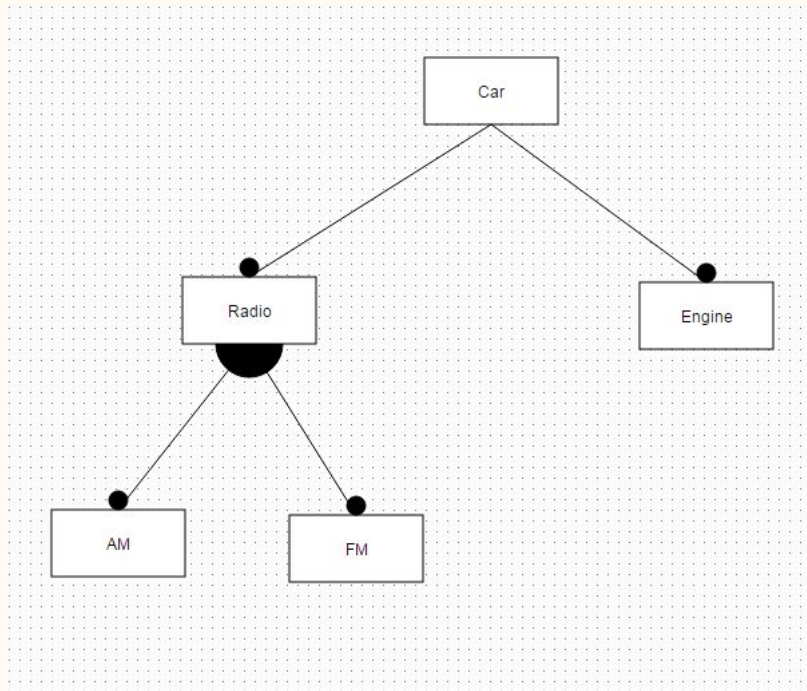
FODA vs GP

Usando regras de composição: Segundo [Kang, 1990] regras de composição definem as semânticas entre *features* que não estão expressas no diagrama.

EX:

- *features* opcionais ou alternativas que **não** podem ser selecionadas **quando** outra *feature* é selecionada: **f1** “mutex-with” **f2**
- *features* que **devem** ser selecionadas **se** uma outra *feature* for selecionada: **f1** “requires” **f2**

Usando regras de composição

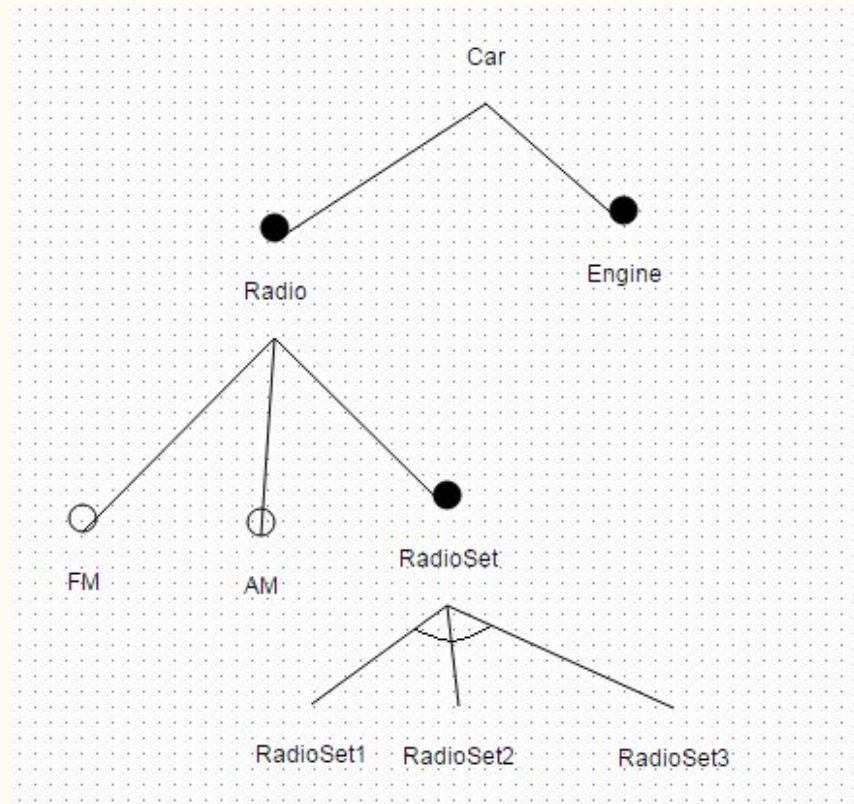


GP Notation

Usando regras de composição

Regras:

- **RadioSet1** “requires” **FM**
- **RadioSet1** “mutex-with” **AM**
- **RadioSet2** “requires” **AM**
- **RadioSet2** “mutex-with” **FM**
- **RadioSet3** “requires” **FM**
- **RadioSet3** “requires” **AM**



FODA Notation

Usando regras de composição

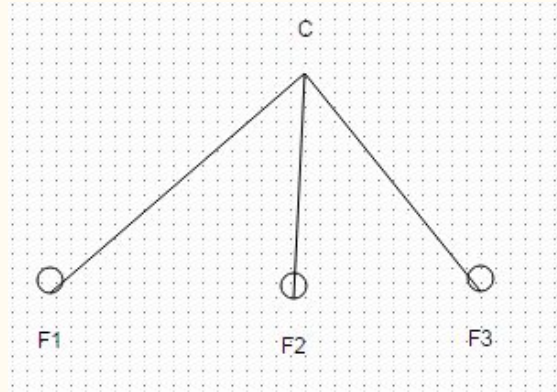
Para saber quantas *subfeatures* precisarão ser criadas basta usar a fórmula:

$N = 2^x - 1$, onde x = número de *or-features* dentro do grupo na notação GP.

Se existem 5 *or-features*, deve ser criado um grupo com 31 *subfeatures* na notação FODA.

FODA vs GP

- *Optional features* e regras de composição.

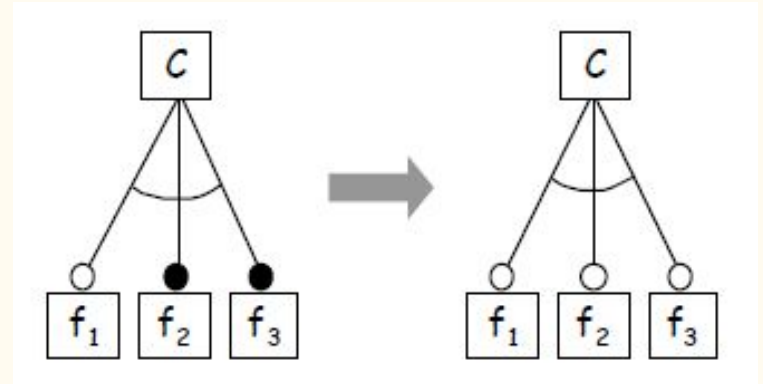


FODA notation

Regras:

- **F1** “mutex-with” **F2** |||| **F1** “mutex-with” **F3**
- **F2** “mutex-with” **F1** |||| **F2** “mutex-with” **F3**
- **F3** “mutex-with” **F1** |||| **F3** “mutex-with” **F1**

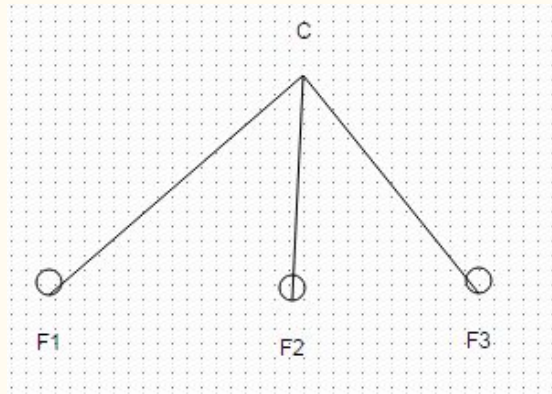
- *Optional alternative features*



Normalizando grupo com *optional XOR-features*

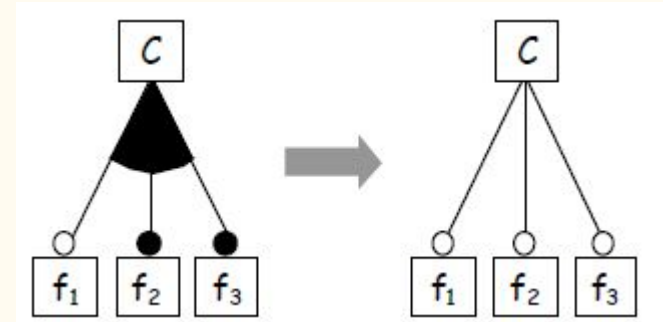
FODA vs GP

- *Optional features*



FODA notation

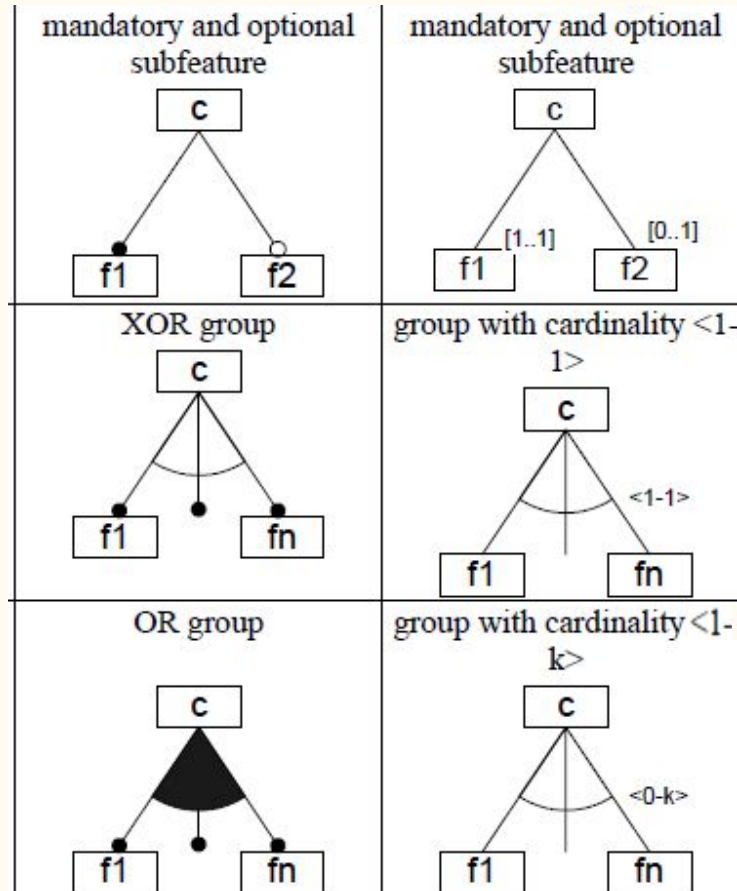
- *Optional Or-features*



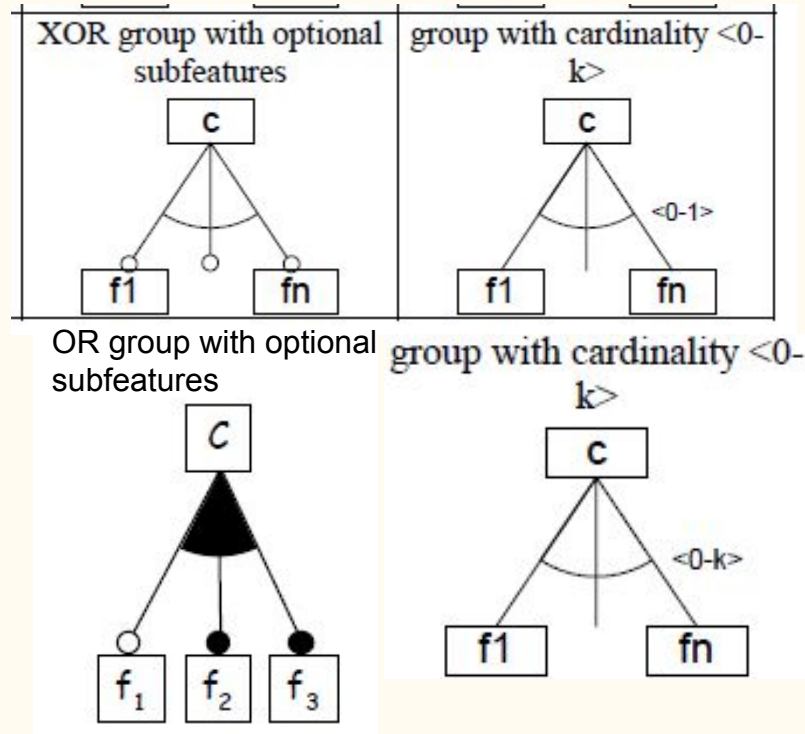
Normalizando grupo com *optional OR-features*

GP vs CB

GP vs CB

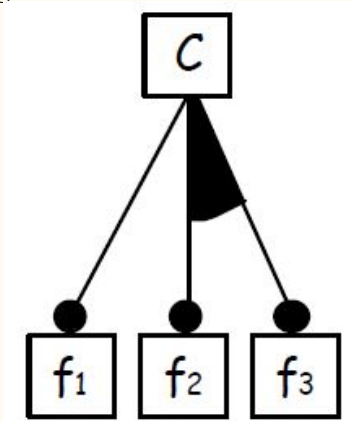


GP vs CB



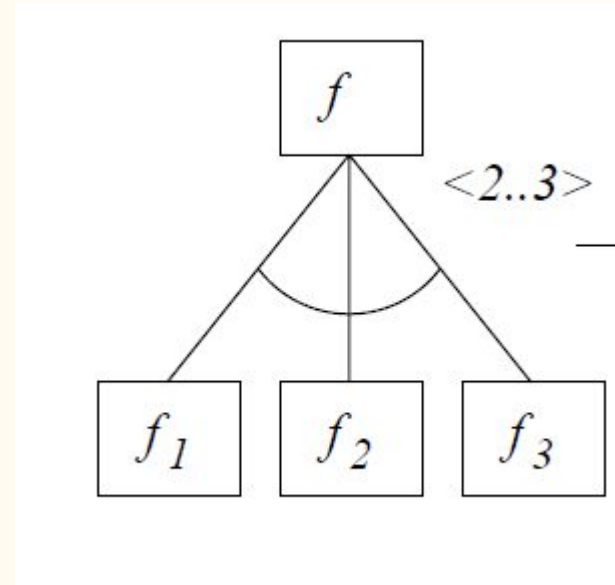
GP vs CB

- Grupo com *or-features*
 - Permite selecionar apenas **uma** *feature*.
- Separar uma das *features* do grupo
 - Limita à **sempre** selecionar a *feature* obrigatória

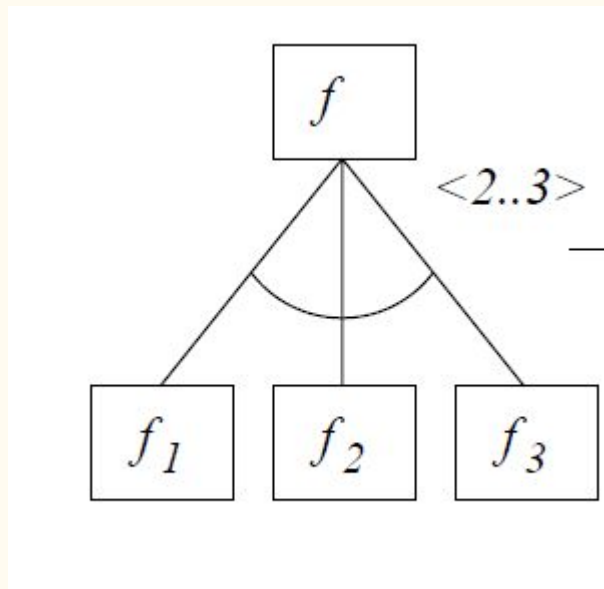


Possível solução: regras de composição.
(Gambiarra?)

- Grupos onde a cardinalidade mínima é maior que 1;

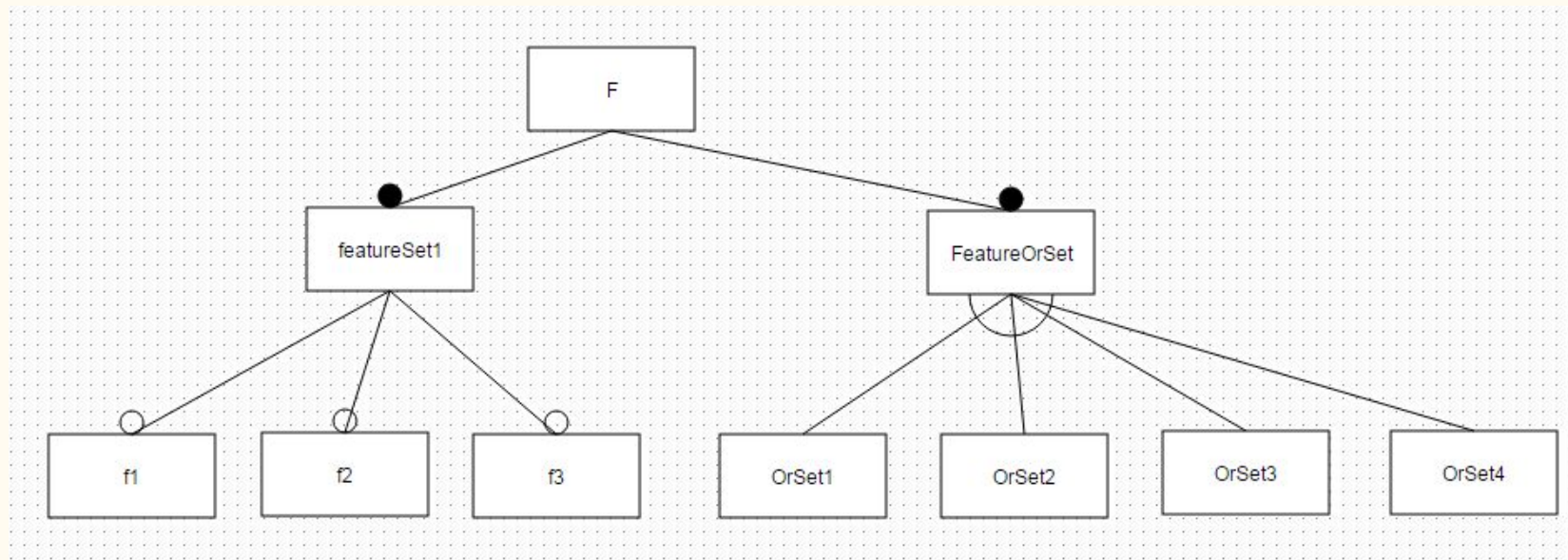


Usando regras de composição



CB notation

Usando regras de composição



GP notation

Usando regras de composição

Regras:

- **OrSet1** “requires” **f1**
- **OrSet1** “requires” **f2**
- **OrSet1** “mutex-with” **f3**
- **OrSet2** “requires” **f1**
- **OrSet2** “requires” **f3**
- **OrSet2** “mutex-with” **f2**
- **OrSet3** “requires” **f2**
- **OrSet3** “requires” **f3**
- **OrSet3** “mutex-with” **f1**
- **OrSet4** “requires” **f1**
- **OrSet4** “requires” **f2**
- **OrSet4** “requires” **f3**

Usando regras de composição

Para saber quantas *subfeatures* precisarão ser criadas basta usar a fórmula:

$$\sum_{i=n}^{n'} \frac{k!}{i! (k-i)!}$$

onde n = cardinalidade mínima, n' = cardinalidade máxima e k = número de *features* do grupo.

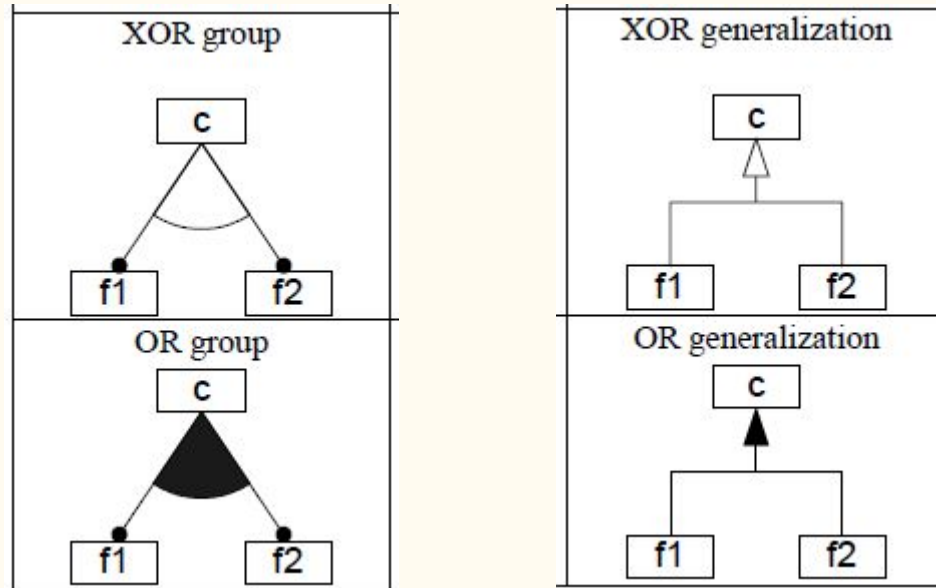
Se existem 4 *features* e a cardinalidade é $\langle 2-3 \rangle$, devemos criar 10 *subfeatures* na notação GP.

FODA vs CB

- A comparação FODA vs CB será idêntica à FODA vs GP.

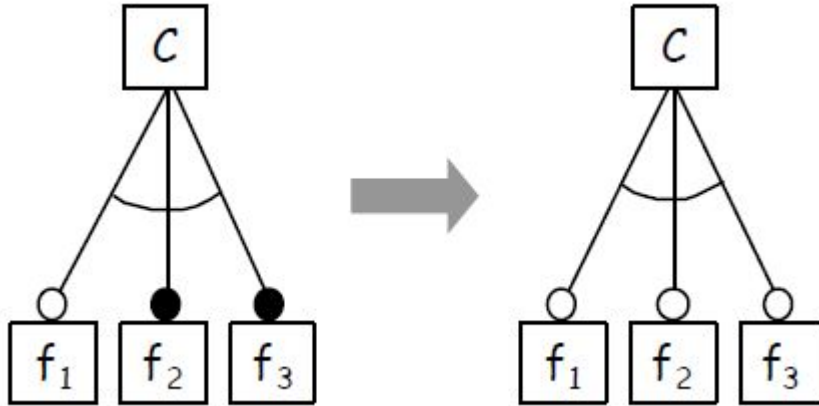
GP vs GBS

GP vs GBS



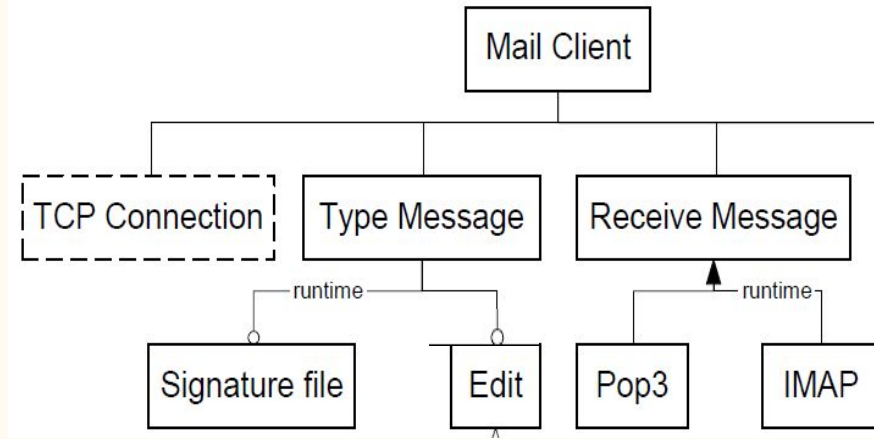
GP vs GBS

- Grupo com *optional XOR-features*
- Normalizado, fica um grupo com *optional alternative features*



Normalizando grupo com *optional XOR-features*

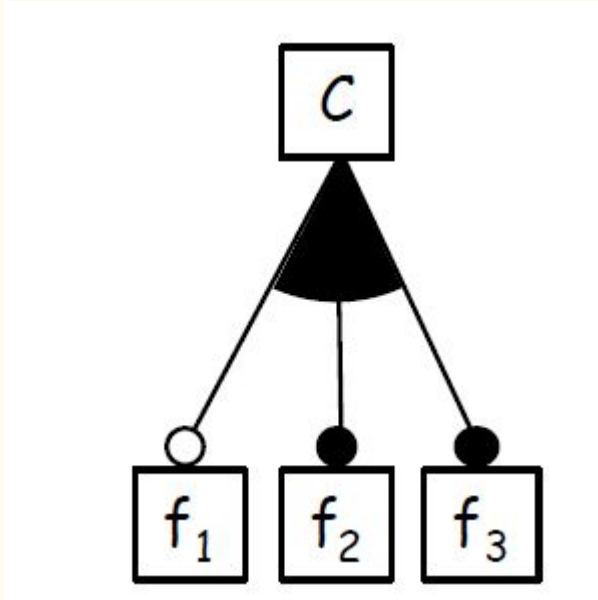
- Criar grupo apenas com *optional features*



Grupo com *optional features* (GBS)

GP vs GBS

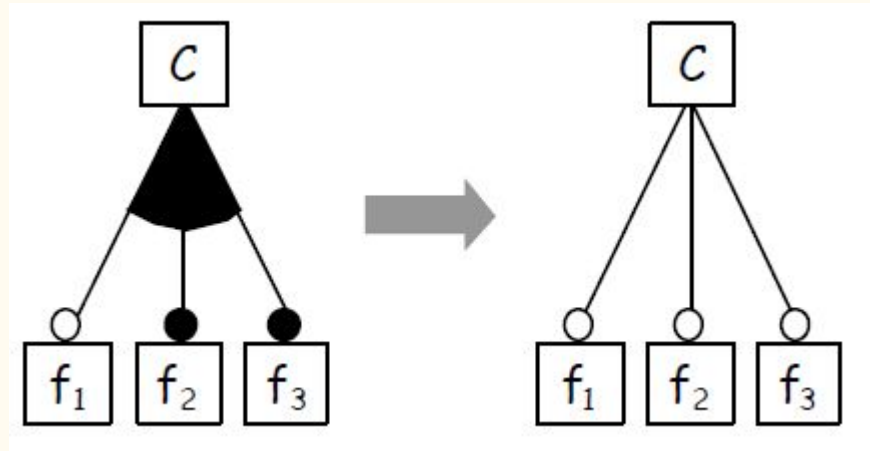
- Grupo com *optional OR-features*



Grupo com *optional OR-features*

- Não é um problema por conta da normalização

Grupo com *optional OR-features*



Normalização de grupo com *optional or-features*

GP vs GBS

Opções:

- Remover *external features*
 - Perda de informação
 - Mante-las como *normal features*
 - Elas se tornam parte do sistema
- *External features*: features fornecidas pela plataforma alvo do sistema. **Não são parte direta do sistema** mas são importantes porque o sistema usa e depende delas.

Possível solução (Gambiarra?): Usar a descrição da *feature* e/ou a *rationale* para identificá-la como não parte direta do sistema.

CB vs GBS

CB vs GBS

- Como GP e CB são equivalentes, a comparação CB vs GBS será **quase** idêntica.
- Já a comparação FODA vs GBS será idêntica à FODA vs GP.

CB vs GBS

Opções:

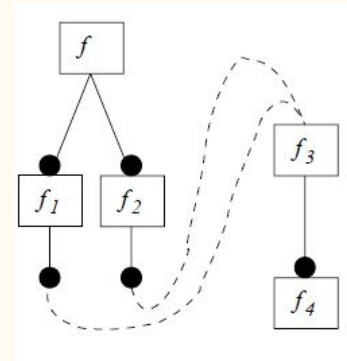
- *External Features*

- Remover *external features*
 - Perda de informação
- Mante-las como *normal features*
 - Elas se tornam parte do sistema

Possível solução (Gambiarra?): Usar a descrição da *feature* e/ou a *rationale* para identificá-la como não parte direta do sistema.

OU

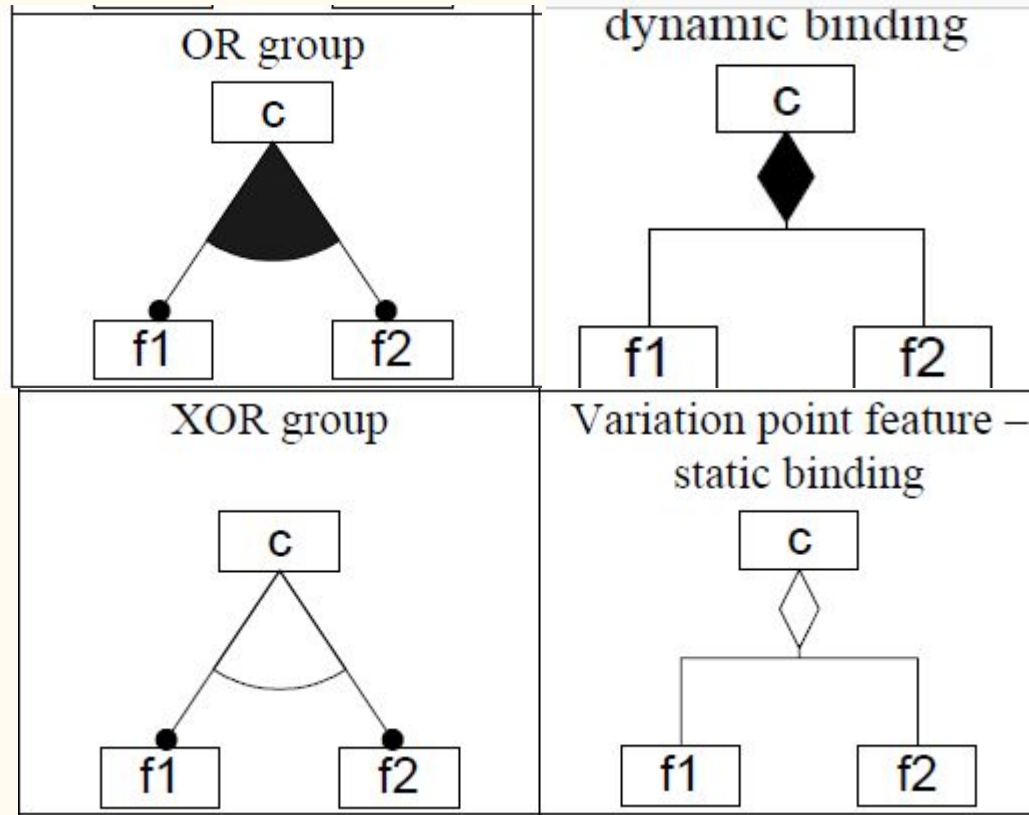
Combinar opção anterior com referência



Referenciando *feature diagram* (CB notation)

GP vs FeatuRSEB

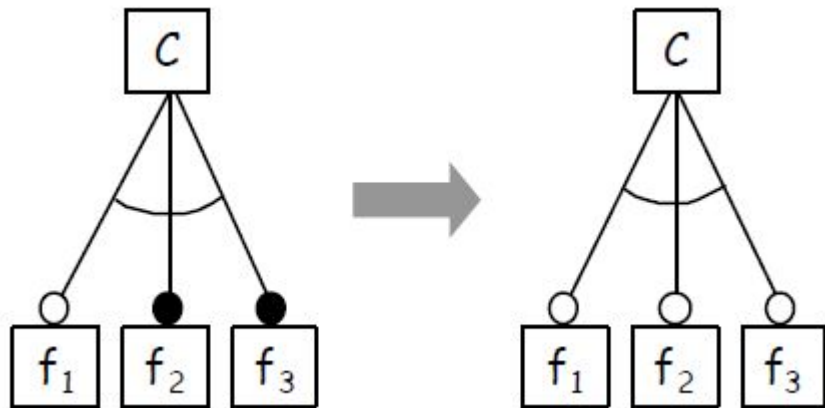
GP vs FeatureRSEB



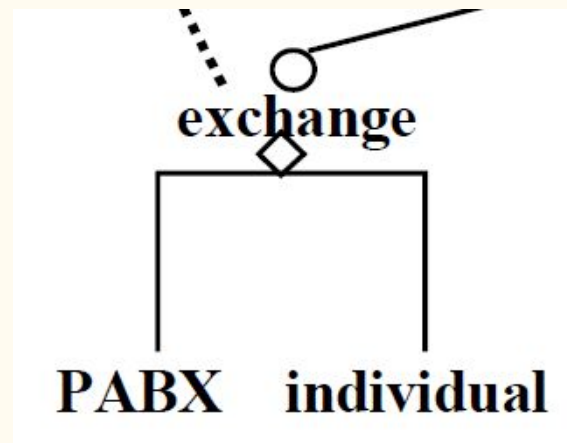
GP vs FeatuRSEB

- Grupo com *optional XOR-features*
- Normalizado, fica um grupo com *optional alternative features*

- Usar *feature* pai como opcional



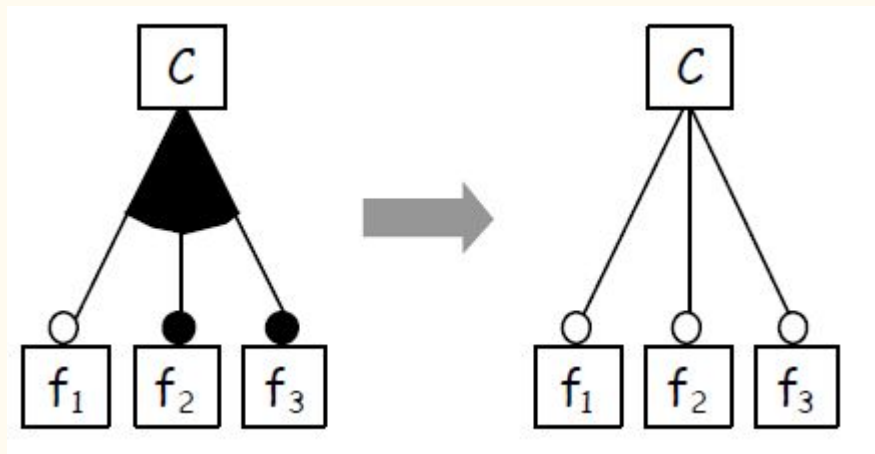
Normalizando grupo com *optional XOR-features*



GP vs FeatuRSEB

- Grupo com *optional OR-features*

- Não é um problema por conta da normalização



Normalização de grupo com *optional or-features*

GP vs FeatureRSEB

- Podemos dizer que GP e FeatureRSEB são equivalentes, fazendo com que as demais comparações com FeatureRSEB sejam semelhantes as comparações à GP.

Referências

Šípka, Miloslav. "Exploring the commonality in feature modeling notations." Proceedings of IIT. SRC (2005): 139-144.

Seidl, Christoph, Tim Winkelmann, and Ina Schaefer. "A Software Product Line of Feature Modeling Notations and Cross-Tree Constraint Languages." Modellierung. 2016.

Kang, K., Cohen, S., Hess, J., Nowak, W. and Peterson, S. (1990). Feature-oriented domain analysis (FODA) feasibility study, Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Griss, Martin L., John Favaro, and Massimo d'Alessandro. "Integrating feature modeling with the RSEB." Software Reuse, 1998. Proceedings. Fifth International Conference on. IEEE, 1998.

Czarnecki, K. and Eisenecker, U. W. (2000). Generative Programming: Methods, Tools, and Applications, AddisonWesley, Boston, MA.

Referências

Czarnecki, K., Helsen, S. and Eisenecker, U. (2005). Formalizing cardinality-based feature models and their specialization, *Software Process Improvement and Practice* 10(1).

Van Gurp, Jilles, Jan Bosch, and Mikael Svahnberg. "On the notion of variability in software product lines." *Software Architecture*, 2001. Proceedings. Working IEEE/IFIP Conference on. IEEE, 2001.