

Sommaire

1. **Introduction à la compilation**
 - Qu'est-ce qu'un compilateur
 - Phases d'un compilateur
 - Outils de construction de compilateurs*(concepts généraux sur le rôle et la structure d'un compilateur)*
2. **Un compilateur simple en une passe**
 - Définition de syntaxe
 - Traduction dirigée par syntaxe
 - Analyse et parsing simple*(exemple d'un compilateur minimal pour illustrer les concepts)*
3. **Analyse lexicale**
 - Rôle de l'analyseur lexical
 - Reconnaissance des tokens
 - Utilisation d'automates finis dans l'analyse lexicale*(mise en place des règles lexicales et reconnaissance des éléments de base)*
4. **Analyse syntaxique**
 - Méthodes de parsing
 - Grammaires et arbres syntaxiques*(construction de l'arbre de syntaxe à partir du code source)*
5. **Traduction dirigée par la syntaxe***(liaison entre la structure syntaxique et la génération de représentations intermédiaires)*
6. **Contrôle de types***(vérification des types et règles de cohérence du langage)*
7. **Environnements d'exécution***(gestion de la mémoire, des variables et de la pile d'exécution)*
8. **Production de code intermédiaire***(génération d'une représentation abstraite avant le code final)*
9. **Production de code***(génération du code machine ou code cible à partir des structures intermédiaires)*
10. **Optimisation de code***(techniques pour améliorer l'efficacité du code généré)*
11. **Écrire un compilateur complet***(conseils pratiques et intégration de toutes les phases)*
12. **Études de cas de compilateurs***(exemples de réalisations réelles ou projets)*
13. **Projets et exercices***(activités pour appliquer les connaissances et améliorer la compréhension)*

